digital

Computer Special Systems

# RH70/Special Massbus Controller

## OPTION DESCRIPTION

ADDENDUM REQUIRED

| | |
|---|---|
| COMPUTER TYPE<br>PDP-11/70 | DRAWING SET NO.<br>RH70-0 |
| PROGRAM NO.<br>N/A | DOCUMENT NO.<br>CSS-MO-F-5.2-27 |
| DATE<br>February 1977 | |

1st Printing May 1976
2nd Printing February 1977



The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

## ILLUSTRATIONS



## TABLES

## PREFACE

This manual describes the operation of the RH70 Massbus controller. The RH70 is a general purpose controller that can be used with any Massbus peripheral and the manual has been written so that it is applicable to any of these peripherals. The terms "Massbus device" and "Massbus drive" are frequently used in the manual to refer to these peripherals. In some cases where it has been necessary to use a specific Massbus device as an example to illustrate a point, the TU45 tape drive has been chosen as an example. For a complete understanding of the operation of the RH70 and other peripheral devices, the user should refer to the operating manual for that peripheral device.

# CHAPTER 1
# RH70/SPECIAL MASSBUS CONTROLLER

## 1.1 GENERAL DESCRIPTION

The RH70/SPECIAL, designed and manufactured by Digital Equipment Corporation, is the standard Massbus Controller for the PDP-11/70 processor. Used in conjunction with any Massbus device, the RH70 provides an extremely fast and reliable mass storage system that can be employed in timesharing or real-time data storage applications. The following major functions are performed by the RH70.

1. Communicates with main memory via Cache in order to fetch and store data.

2. Communicates with the control processor via the Unibus in order to receive commands, provide error and status information, and generate interrupts.

3. Interfaces with from one to eight drives via the Massbus.

The RH70, which can accommodate up to eight standard Massbus devices, is housed in prewired module slots in the PDP-11/70 backplane. As many as four RH70s may be used on one PDP-11/70. Figure 1-1 is a simplified block diagram of a system incorporating the use of an RH70.

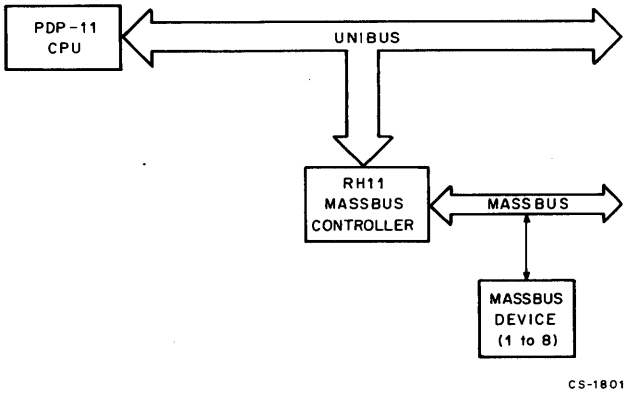



Figure 1-1 RH70 Simplified System Diagram

## 1.2 OPERATION

The RH70 has a Unibus port to allow access to the control and status registers in the RH70 and in the Massbus device. Data transfer takes place over a 32-bit wide data bus into main memory through the Cache memory control.

## 1.3 SPECIFICATIONS*

**Mechanical**

Prerequisites

Mounting space for the RH70 controller in the PDP-11/70 backplane.

RH70 Logic

3 quad-height modules
1 hex-height module
3 double-height modules

CPU Backplane

4 module slots

**Electrical**

Input Power

+5.0 ± 0.25 Vdc at 8.5 A max.
−15.0 ± 1.5 Vdc at 0.5 A max.

Power Supplies

Uses standard power supply and regulators of PDP-11/70 CPU

Logic Levels

TTL (Massbus transceivers have differential outputs)

Module Type

M-Series

**Operational**

Transfer Rate

The transfer rate of the RH70 is primarily a function of the available memory bandwidth and the amount of other memory activity within the system.

Output Levels

Differential, maximum total length of the Massbus cabling including any cabling within the Massbus device is 120 feet.

*Specifications are subject to change without notice

**Environmental**

Temperature

32° – 122° F (0° – 50° C) Class C

Relative Humidity

8% to 90%, no condensation

Vibration Shock

1.89 G rms, 10 – 300 Hz

20 G, half sine, 30 ms duration, any plane

## 1.4 RELATED DOCUMENTATION

| Title | Document Number |
|---|---|
| TWU45 Magtape System Manual | CSS-MO-F-5.2-25 |
| TM02-FE/FF Tape Drive Option Description | CSS-MO-F-5.2-22 |
| Magtape Transport Adapter Option Description TU45 Tape Transport Manual | CSS-MO-F-5.2-23 |
| PDP-11 Peripherals Handbook | 2002, 20175.4526 |
| PDP-11/70 Processor Handbook | EB-04588/750100 |
| KB11-B Processor System (PDP-11/70 Manual) | 19H161 |
| PDP-11/70 Maintenance and Installation Manual | EX-11070 MM |

# CHAPTER 2
# MASSBUS INTERFACE

## 2.1 GENERAL

The Massbus provides the interface between the RH70 Controller and the Massbus Device. The Massbus can be up to 120 ft in length and up to eight drives may be connected in a daisy-chain configuration. The Massbus consists of two sections: a Data Bus and a Control Bus section. These buses are described in the following paragraphs.

## 2.2 DATA BUS

The Data Bus section of the Massbus consists of a 19-bit (18 data bits, plus parity bit) parallel data path and six control lines (Figure 2-1).

*Parallel Data Path* – The parallel data path consists of an 18-bit data path, designated D00 – D17, and an associated parity bit (DPA). The data path is bidirectional and employs odd parity. Data is transmitted synchronously, using a clock generated in the drive.

*RUN* – After a data transfer command has been written into the control register of a drive, the drive connects to the data bus. The controller then asserts the RUN line to initiate the function.

*Occupied (OCC)* – This signal is generated by the drive to indicate "data bus busy." As soon as a valid data transfer command is written into a drive, and the command is accepted, the drive asserts OCC. Various errors may cause a drive to be unable to execute a command. The controller will time out in these cases, due to no assertion of OCC or SCLK, and the MXF (Missed Transfer Error) will be set in the controller. OCC is negated at the trailing edge of the last EBL pulse of a transfer.

*End-of-Block (EBL)* – This signal is asserted by the drive at the end of each record (after the last SCLK pulse). For certain error conditions, where it is necessary to terminate operations immediately, EBL is asserted prior to the normal time for the last SCLK. The data transfer is terminated prior to the end of the record in this case.

*Exception (EXC)* – This signal is asserted when an abnormal condition occurs in the drive. The drive asserts this signal to indicate an error during a data transfer command (Read, Write, or Write-check). Exception is asserted at, or prior to, assertion of EBL and is negated at the negation of EBL.

*Sync CLK (SCLK), Write CLK (WCLK)* – These signals are the timing signals used to strobe the data in the controller and/or in the drive. During a read operation, the RH70 strobes the data lines on the negation of SCLK and the drive changes the data on the assertion of SCLK. During a write operation, the controller receives SCLK and echoes it back to the drive as WCLK. On the assertion of WCLK, the drive strobes the data lines and on the negation of WCLK, the controller changes the data on the data lines.

## 2.3 CONTROL BUS

The Control Bus section of the Massbus consists of a 17-bit (16 bits, plus parity) parallel control and status data path, and 14 control lines (Figure 2-1).

*Parallel Control and Status Path* – The parallel control and status path consists of a 16-bit parallel data path, designated C00 – C15, and an associated parity bit (CPA). The control and status lines are bidirectional and employ odd parity.

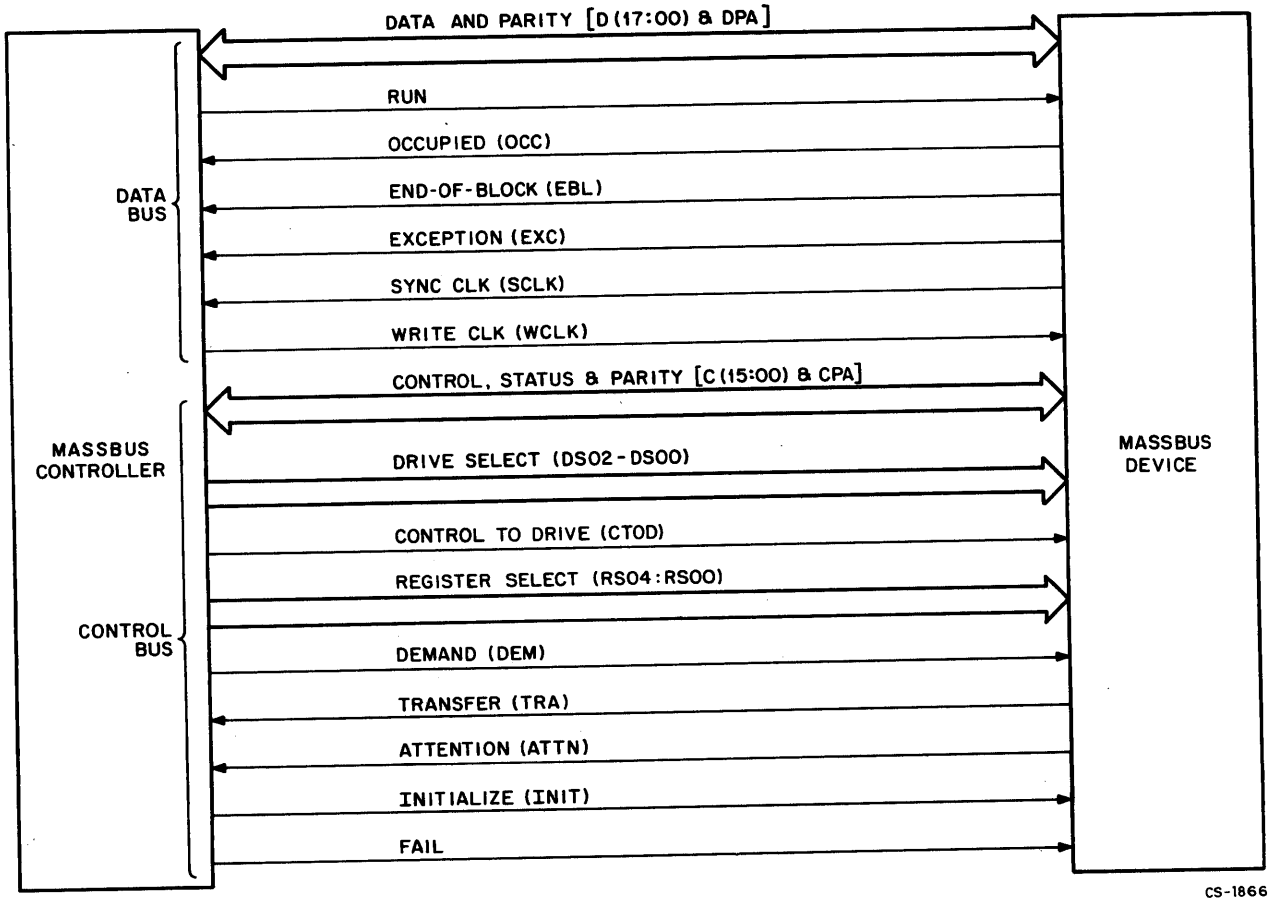



Figure 2-1 Massbus Interface Lines

*Drive Select, DS (2:0)* – These three lines transmit a 3-bit binary code from the controller to select a particular drive. The drive is linked to the Control Bus when the (unit) select number in the drive corresponds to the transmitted binary code.

*Controller-to-Drive (CTOD)* – This signal is generated by the controller and indicates the direction in which control and status information is to be transferred. For a controller-to-drive transfer, the controller asserts CTOD. For a drive-to-controller transfer, the controller negates this signal.

*Register Select, RS (4:0)* – These five lines transmit a 5-bit binary code from the controller to the selected drive. The binary code selects one of the drive registers.

**NOTE**

**It is possible to have as many as 32 registers in a Massbus device. The number of registers that are actually implemented, the function of these registers, and their Unibus addresses will vary from one device to the next. In the case of the TU45/TM02 there are ten registers (including the MTCS1 register) present in the TM02 Tape Controller, designated by codes 00 – 11. If a register code higher than 11 is selected, an Illegal Register (ILR) Error occurs.**

Corresponds To:

| Massbus Address | Register Name | Register Mnemonic | Unibus Address** (Octal) |
|---|---|---|---|
| 00 | Control & Status 1 Register* | MTCS1 | 772440 |
| 01 | Drive Status | MTDS | 772452 |
| 02 | Error | MTER | 772454 |
| 03 | Maintenance | MTMR | 772464 |
| 04 | Attention Summary | MTAS | 772456 |
| 05 | Frame Count | MTFC | 772446 |
| 06 | Drive Type | MTDT | 772466 |
| 07 | Check Character | MTCK | 772460 |
| 10 | Serial Number | MTSN | 772470 |
| 11 | Tape Control | MTTC | 772472 |

* This register is shared by the RH70 and the TM02 controller.

The RH70 Massbus Controller Registers are:

| Register Name | Register Mnemonic | Unibus Address** (Octal) |
|---|---|---|
| Control & Status 1 | MTCS1 | 772440 |
| Word Count | MTWC | 772442 |
| Bus Address | MTBA | 772444 |
| Control and Status 2 | MTCS2 | 772450 |
| Data Buffer | MTDB | 772462 |
| Bus Address Extension | MTBAE | 772474 |
| Control and Status 3 | MTCS3 | 772476 |

*Demand (DEM)* – This signal is asserted by the controller to indicate that a transfer is to take place on the Control Bus. For a controller-to-drive transfer, DEM is asserted by the controller when data is present and settled on the Control Bus. For a drive-to-controller transfer, DEM is asserted by the controller to request data and is negated when the data has been strobed off the Control Bus. In both cases, the RS, DS, and CTOD lines are generated and allowed to settle before assertion of DEM.

*Transfer (TRA)* – This signal is asserted by the drive in response to DEM. For a controller-to-drive transfer, TRA is asserted when the data is strobed and is negated when DEM is negated. For a drive-to-controller transfer, TRA is asserted after the data is asserted on the bus and negated when the negation of DEM is received.

*Attention (ATTN)* – This line is shared by all eight drives attached to a controller; it may be asserted by any drive as a result of an abnormal condition or status change in the drive. An ATA status bit in each drive is set whenever that drive is asserting the ATTN line. ATTN may be asserted due to any of the following conditions:

1. An error while no data transfer is taking place (asserted immediately).
2. Completion of a data transfer command if an error occurred during a data transfer (asserted at the end of the transfer).
3. Completion of a non-data transfer command (such as a space).

ndard address assignment for the TU45, for address ions pertaining to other Massbus devices refer to that device's accompanying manual.

The ATA bit in a drive may be cleared by the following actions:

1. Asserting INIT on the Massbus (affects all eight drives).

2. Writing a 1 into the Attention Summary register (in the bit position for this drive). This clears the ATA bit; however, it does not clear the error.

3. Writing a valid command into the Control register (with the GO bit set). Note that clearing the ATA bit of one drive does not always cause the ATTN line to be negated because other drives may also be asserting the line.

NOTE

**There are three cases in which ATA is not reset when a command is written into the Control register (with the GO bit set): 1) if there is a Control Bus parity error on the write, 2) if an error was previously set, or 3) if an illegal function code (ILF) is written.**

*Initialize (INIT)* – This signal is asserted by the controller to perform a system reset of all drives. It is asserted when a 1 is written into the CLR bit (bit 05 of MTCS2) or when Unibus INIT is asserted on the Unibus. When a drive receives the INIT pulse, it immediately aborts the execution of any current command and performs all actions described for the Drive Clear command.

*Fail* – When asserted, this signal indicates that a power-fail condition has occurred in the controller. In particular, the drive inhibits reception of the INIT and DEM signals at the drive.

## 2.4 COMMAND INITIATION

To initiate a command in a drive via the Massbus, the controller (or the central processor via the controller) writes a word in the MTCS1 register which causes a word to be written into the drive's Control register (00). The word contains a command function code in bits 05 – 01 and a GO bit in bit 00. The GO bit is set when initiating a command. If the command specified is valid, the drive which has been addressed by the program executes the command.

Commands are of two types: non-data transfer commands (such as Drive Clear, space) and data transfer commands (such as Read, Write, and Write-check). The command function code bits (05 – 00, including GO in MTCS1) are $01_8$ – $47_8$ for non-data transfer commands and $51_8$ – $77_8$ for data transfer commands.

### 2.4.1 Non-Data Transfer Commands

Non-data transfer commands have effect only on the state of the drive. The controller merely writes the command word (with GO bit set) into the drive's Control register. At the completion of the command execution, the drive typically asserts the ATTN line to signal its completion.

If the non-data transfer command code written into the drive is not recognized by the drive as a valid command, the drive will immediately signal an error by asserting the ATTN line. The Illegal Function Error (ILF) is set.

### 2.4.2 Data Transfer Commands

When any data transfer command code (with the GO bit set) is written into the drive's Control register, the controller expects data transfer on the Data Bus to begin thereafter. The controller resets its RDY (Controller Ready) bit as soon as the data transfer command code is written into a drive. The drive normally responds by asserting the OCC line. The controller asserts RUN and then data is transferred to or from the specified drive.

If an error occurs in a drive during a data transfer command, the drive asserts the EXC line. This line remains asserted until the trailing edge of the last EBL pulse. The RH70 Controller always negates the RUN line when it detects EXC asserted, so that data transfer is terminated at the end of the record in which the error was signaled.

## 2.5 READING/WRITING DRIVE REGISTERS

The process of reading/writing drive registers is accomplished via the asynchronous (Control Bus) portion of the Massbus (Figure 2-1). The RH70 initiates the action by selecting a drive (DS2–DS0), selecting a register (RS4–RS0) in that drive, selecting a direction of transfer (CTOD), and either reading–writing the register via the 17 bidirectional control lines [C (15:00) and CPA]. After a deskew delay to allow the control lines to stabilize, the RH70 asserts DEM. Upon receiving the DEM assertion, the drive checks the CTOD line to ascertain whether a read or write is to occur.

If a register read operation is specified, the drive will gate the contents of the specified register onto the Control Bus and issue TRA. When the RH70 receives TRA, it will gate the control lines onto the Unibus. After a deskew delay, the RH70 asserts SSYN to the processor. When the processor receives the control data and SSYN, it clears MSYN. The clearing of MSYN negates SSYN and DEM. The negation of DEM causes TRA to be negated and completes the operation.

If a register write operation is specified, the RH70 gates the control data onto the Control Bus when it issues DEM. The drive will transfer the data from the Control Bus into the specified drive register and assert TRA, which causes the assertion of SSYN in the RH70. SSYN is transferred to the processor and causes MSYN to be cleared. The clearing of MSYN causes SSYN and DEM to be negated. The negation of DEM causes TRA to be negated to complete the operation.

The Massbus structure allows a register read operation (asynchronous Control Bus) to occur while a data transfer (synchronous Data Bus) is taking place. Any attempt by the RH70 to write a register in a drive performing a data transfer operation (except for the Maintenance and Attention Summary registers) will cause the drive to set the Register Modification Refused (RMR) error bit (Chapter 4).

## 2.6 MASSBUS PHYSICAL DESCRIPTION

The Massbus consists of 56 signals including data, control, status, and parity. These signals are routed between the RH70 and the drives by three 40-conductor flat cables. Since Massbus signal transmission (with exception of the FAIL signal) is accomplished by differential transmitter/receiver pairs, each cable can accommodate 20 differential signals.

On the drive end, the cables are plugged into M5903 Massbus Drive Transceiver modules. The last drive has either M5903-YA modules or M5903 modules with H870 mini-terminators which terminate the bus. On the controller end, each cable plugs into an M5904 Massbus Controller Transceiver module (described in subsequent paragraphs). Each M5904 module in turn plugs into a slot (see Figure 2-1) in the 11/70 backplane to complete the signal path.

Table 2-1 shows the Massbus signals and their associated pin assignments.

**NOTE**

**Massbus cables are to be installed per markings on the cables.**

**Table 2-1**
**Massbus Signal Cable Designations**

| Cable | Pin* | | Polarity | Designation |
|---|---|---|---|---|
| Massbus Cable A | A | 1 | - | MASS D00 |
| | B | 2 | + | |
| | C | 3 | + | MASS D01 |
| | D | 4 | - | |
| | E | 5 | - | MASS D02 |
| | F | 6 | + | |
| | H | 7 | + | MASS D03 |
| | J | 8 | - | |
| | K | 9 | - | MASS D04 |
| | L | 10 | + | |
| | M | 11 | + | MASS D05 |
| | N | 12 | - | |
| | P | 13 | - | MASS C00 |
| | R | 14 | + | |
| | S | 15 | + | MASS C01 |
| | T | 16 | - | |
| | U | 17 | - | MASS C02 |
| | V | 18 | + | |
| | W | 19 | + | MASS C03 |
| | X | 20 | - | |
| | Y | 21 | - | MASS C04 |
| | Z | 22 | + | |
| | AA | 23 | + | MASS C05 |
| | BB | 24 | - | |
| | CC | 25 | - | MASS SCLK |
| | DD | 26 | + | |
| | EE | 27 | + | MASS RS3 |
| | FF | 28 | - | |
| | HH | 29 | + | MASS ATTN |
| | JJ | 30 | - | |
| | KK | 31 | - | MASS RS4 |
| | LL | 32 | + | |
| | MM | 33 | - | MASS CTOD |
| | NN | 34 | + | |
| | PP | 35 | + | MASS WCLK |
| | RR | 36 | - | |
| | SS | 37 | + | MASS RUN |
| | TT | 38 | - | |
| | UU | 39 | | SPARE |
| | VV | 40 | | GND |

*Alternate pin designation schemes

**Table 2-1 (Cont)**
**Massbus Signal Cable Designations**

| Cable | Pin* | Polarity | Designation |
|---|---|---|---|
| Massbus Cable B | A 1 | - | MASS D06 |
| | B 2 | + | |
| | C 3 | + | MASS D07 |
| | D 4 | - | |
| | E 5 | - | MASS D08 |
| | F 6 | + | |
| | H 7 | + | MASS D09 |
| | J 8 | - | |
| | K 9 | - | MASS D10 |
| | L 10 | + | |
| | M 11 | + | MASS D11 |
| | N 12 | - | |
| | P 13 | - | MASS C06 |
| | R 14 | + | |
| | S 15 | + | MASS C07 |
| | T 16 | - | |
| | U 17 | - | MASS C08 |
| | V 18 | + | |
| | W 19 | + | MASS C09 |
| | X 20 | - | |
| | Y 21 | - | MASS C10 |
| | Z 22 | + | |
| | AA 23 | + | MASS C11 |
| | BB 24 | - | |
| | CC 25 | - | MASS EXC |
| | DD 26 | + | |
| | EE 27 | + | MASS RS0 |
| | FF 28 | - | |
| | HH 29 | + | MASS EBL |
| | JJ 30 | - | |
| | KK 31 | - | MASS RS1 |
| | LL 32 | + | |
| | MM 33 | - | MASS RS2 |
| | NN 34 | + | |
| | PP 35 | + | MASS INIT |
| | RR 36 | - | |
| | SS 37 | + | MASS SP1 |
| | TT 38 | - | |
| | UU 39 | | SPARE |
| | VV 40 | | GND |

*Alternate pin designation schemes

**Table 2-1 (Cont)**
**Massbus Signal Cable Designations**

| Cable | Pin* | Polarity | Designation |
|---|---|---|---|
| Massbus Cable C | A 1 | - | MASS D12 |
| | B 2 | + | |
| | C 3 | + | MASS D13 |
| | D 4 | - | |
| | E 5 | - | MASS D14 |
| | F 6 | + | |
| | H 7 | + | MASS D15 |
| | J 8 | - | |
| | K 9 | - | MASS D16 |
| | L 10 | + | |
| | M 11 | + | MASS D17 |
| | N 12 | - | |
| | P 13 | - | MASS DPA |
| | R 14 | + | |
| | S 15 | + | MASS C12 |
| | T 16 | - | |
| | U 17 | - | MASS C13 |
| | V 18 | + | |
| | W 19 | + | MASS C14 |
| | X 20 | - | |
| | Y 21 | - | MASS C15 |
| | Z 22 | + | |
| | AA 23 | + | MASS CPA |
| | BB 24 | - | |
| | CC 25 | - | MASS OCC |
| | DD 26 | + | |
| | EE 27 | + | MASS DS0 |
| | FF 28 | - | |
| | HH 29 | + | MASS TRA |
| | JJ 30 | - | |
| | KK 31 | - | MASS DS1 |
| | LL 32 | + | |
| | MM 33 | - | MASS DS2 |
| | NN 34 | + | |
| | PP 35 | + | MASS DEM |
| | RR 36 | - | |
| | SS 37 | + | MASS SP2 |
| | TT 38 | - | |
| | UU 39 | H | MASS FAIL |
| | VV 40 | | GND |

*Alternate pin designation schemes

# CHAPTER 3
# THEORY OF OPERATION

## 3.1 GENERAL

This chapter describes the theory of operation of the RH70 Controller in two functional groupings – the Register Control Path and the Data Path. These are described in detail in the following paragraphs.

## 3.2 REGISTER ACCESS CONTROL PATH

The Register Control Path provides the interface that enables the program to read from or write into any register in the RH70 or associated drive. Specific bits in these registers are designated as follows:

*Read only* bits indicate that the program can read the status of these bits but cannot load them; *write only* bits indicate that the program can load them but will read back a 0; *read/write* bits indicate that the program may load them and read back the status.

The RH70 examines Unibus address bits 17 – 05 (17 – 06 if there are a total of more than 16 registers) to determine if the register being addressed is an RH70 register (Figure 3-1). The address field can be defined by a set of jumpers within the RH70. The Unibus address is compared with the set of jumpers and if the two match, the addressed register is a valid RH70 register, which enables the circuitry for a register function. If the Unibus address does not compare with the jumpers, the RH70 will not accept the address and will not initiate a data transfer with the processor.

The RH70 contains two registers (Bus Address Extension and Control and Status 3 registers) which float in the Massbus address field. Their respective addresses may vary in different subsystems, depending on the number of registers within a given system. Consequently, the registers are logically defined as being the last two registers in a system. The number of registers is classified as the *register length field.*

The logic implements this in the following manner. Address bits 4 – 2 from the Unibus are compared to the register length field consisting of a set of jumpers weighted 2, 4, 8, and 16.

**NOTE**

**If the subsystem contains more than 16 registers, address bit 5 from the Unibus is also compared to the jumpers selected in the register length field.**

The Unibus address bits and the register length field jumpers are compared in a comparator which yields the following outputs:

A>B – denotes that the register length field is greater than the Unibus address specified, indicating that a valid register has been addressed. If the register is remote, a Massbus cycle is initiated.

A<B – denotes that the Unibus address is greater than the register length field, indicating that an illegal register has been addressed.

A=B – indicates that either the CS3 or BAE register has been specified. If address bit 01 is asserted, the CS3 register is specified; if the bit is unasserted, the BAE register is specified. For this case, no Massbus cycle is initiated.

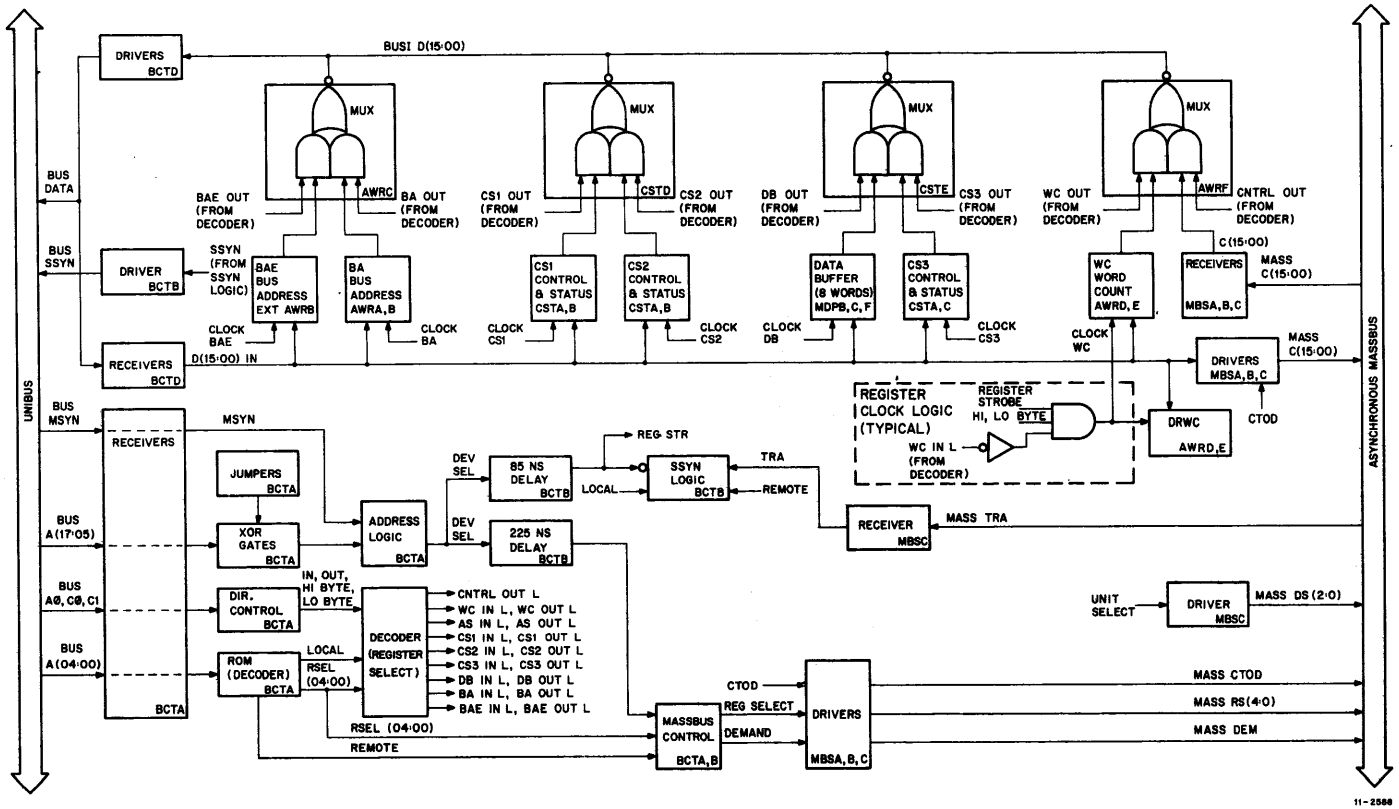



Figure 3-1 Register Control Path

### 3.2.1 Writing a Local Register

Unibus address bits 04 – 01 (05 – 01 if more than 16 registers are employed) select a cell in a Read Only Memory (ROM) which specifies a unique register. The ROM outputs are register select signals (RSEL 04:00), two coded bits (M6 and M7), and a LOCAL REG H signal. Since this description involves accessing a local register (one contained in the RH70), LOCAL is generated at the output of the ROM as LOCAL REG H. When this signal is unasserted or low and a register operation is being performed, a remote register (unless CS3 or BAE) is selected. Signals RSEL 01:00 and M6 and M7 are supplied to the register decoders to select one of the local registers. RSEL 04:00 is also supplied to the Massbus control logic, but is inhibited from the Massbus because a "write local register" function is specified and REMOTE REG remains unasserted.

Unibus control lines A0, C0, and C1 specify the direction of transfer and also specify byte or word addressing. When writing a register, the C0 and C1 lines are encoded for a DATO or DATOB (if byte addressing is specified). The A0, C0, and C1 control lines are supplied to a direction control network which generates IN, OUT, HI BYTE, or LO BYTE signals, depending on the cycle desired. These signals are fed to the register decoder where they are used in decoding the various register enable signals.

The Unibus data lines are connected to the RH70 and contain the data used to load the desired register.

When BUS MSYN is received from the central processor (150 ns after the data, control and address are placed on the Unibus), a DEV SEL (Device Select) signal is generated, which enables the register decoder to generate the appropriate enable signal for the register specified. Signal REG STR is created 85 ns later and is ANDed with the HI BYTE or LO BYTE signal and the specified register enable signal from the register decoder. The signals designated IN are used for writing local registers; the signals designated OUT are used for reading local registers. For example, if it was desired to write into the WC (Word Count) register, the register decoders specify the WC IN L signal, which is ANDed with HI BYTE or LO BYTE and REG STR to generate a clock used to load the WC register. The data is clocked into the WC register at the time of REG STR. The trailing edge of this signal, which is 150 ns long, causes SSYN to be asserted. The central processor receives SSYN and lowers MSYN, which deselects the RH70 from the Unibus. The lowering of MSYN then causes SSYN to be lowered, and 75 ns after the lowering of MSYN, the address lines change and the cycle is completed.

### 3.2.2 Reading a Local Register

The process of reading a local register is the same as that described for writing a local register (Paragraph 3.2.1) with the following exceptions:

1. The C0 and C1 Unibus control lines are decoded for a DATI or DATIP operation.

2. When reading a local register, the register "OUT" signals of the register select decoders gate the contents of the register on the BUSI lines for transmission to the processor via the Unibus.

### 3.2.3 Writing a Remote Register (Figure 3-2)

A remote register is defined as a register located in the drive. The data path for writing a remote register is from the Unibus via data lines D00–D15 IN H on to the Massbus, via control lines MASS C00–C15 H, where the data is received by the selected drive and loaded into the specified register in that drive. A CTOD (Controller to Drive) signal on the Massbus specifies the direction of transfer to the drive.

The upper address bits of the Unibus address are compared with a set of jumpers in the RH70 to enable the register selection logic, previously described. Unibus address bits 04 – 01 (05 – 01 if more than 16 registers are employed in the system) select a cell in the ROM (Read Only Memory) which specifies a particular register. The outputs of the ROM are register select signals RSEL (04:00), two coded bits (M6 and M7), and the LOCAL REG signal. The selected drive, whose unit number was preloaded by the programmer in the CS2 register, is specified by device select lines DS00 – DS02 on the Massbus.

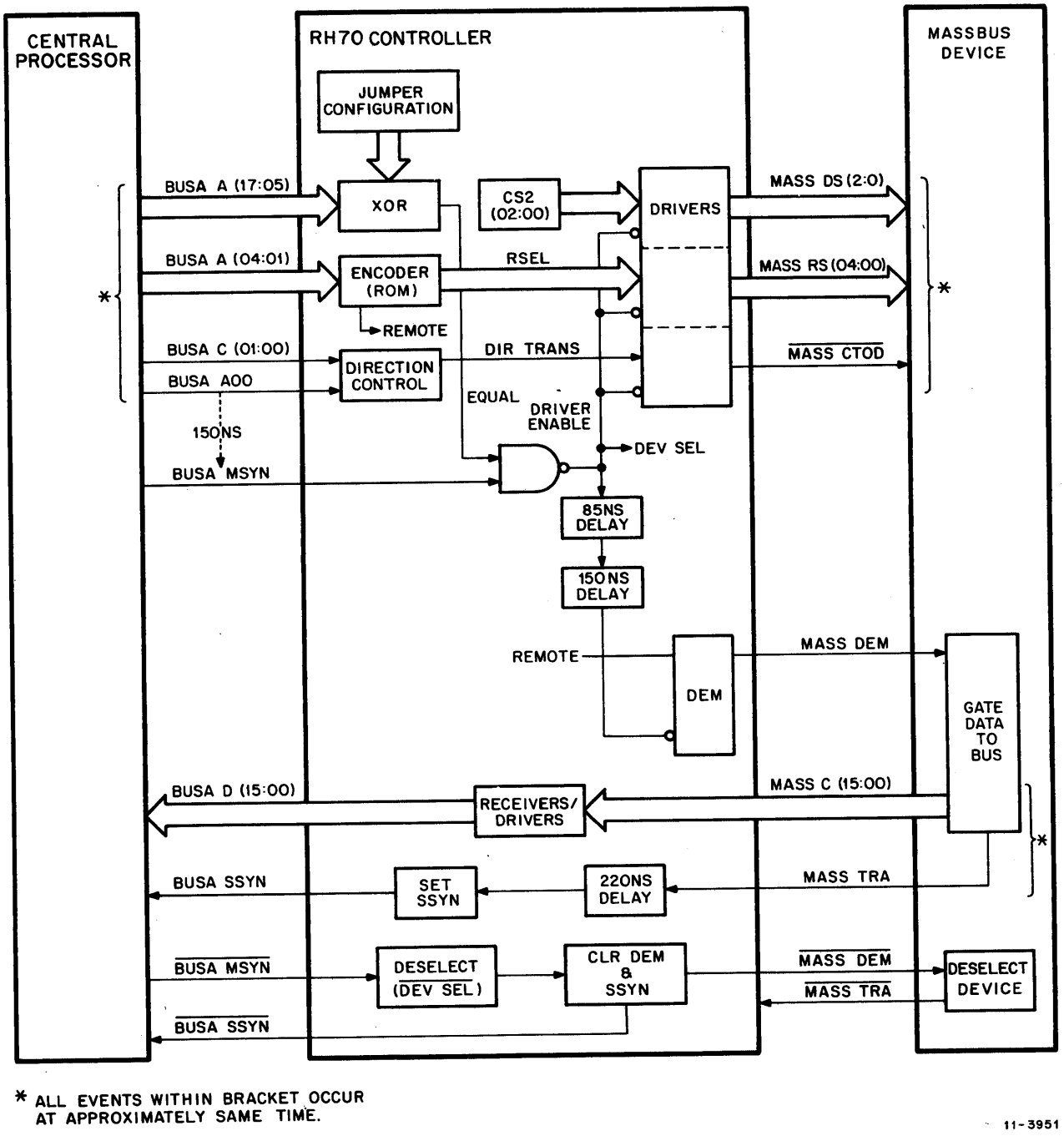



Figure 3-2 Writing Remote Register Interface

When the MSYN signal is received over the Unibus by the RH70, the DEV SEL signal is enabled and a delay of 235 ns occurs before the RH70 issues DEM to the Massbus. This delay allows the select and data lines to settle and be decoded on the Massbus before the drive strobes the Massbus control lines. When the drive receives DEM and recognizes the unit address as its own, and when the data has been clocked into the appropriate drive register, it issues transfer (TRA) to the RH70. When the RH70 receives TRA, indicating that the drive has obtained the data, it issues SSYN to the processor. SSYN signals the processor that the slave (RH70) has finished the cycle, and the processor removes MSYN, which in turn causes SSYN to go unasserted. Also, MSYN going unasserted, removes the DEV SEL signal which causes DEM to drop. This action in turn causes TRA from the drive to go unasserted. The address and data is then removed from the Unibus and Massbus to complete the cycle.

### 3.2.4 Reading a Remote Register (Figure 3-3)

The process of reading a remote register is similar to that of writing a remote register with the following exceptions:

1. The data path for reading a remote register is from the drive to Massbus control lines C00 - C15 H, to the RH70 open-collector multiplexers (8234), to the BUSI lines, and then to the Unibus data lines D00–D15.

2. Upon receipt of TRA when writing a remote register, SSYN is immediately sent to the CPU. When reading a remote register, however, SSYN is delayed 220 ns from TRA to ensure that the data is present and settled on the Unibus.

## 3.3 DMA DATA PATH

The DMA data path is used for memory-to-Massbus data transfers or from Massbus-to-memory data transfers. The data path consists of an eight-word data buffer and associated data buffer control logic. The data buffer provides for temporary storage of up to eight data words. It is designed to accept or provide memory with single- (16 bits) or double-word (32 bits) data and to convert double words to single words for transfer to the Massbus. Parity checking and generating is also provided for Massbus and memory data words.

Due to the complexity of the Data Buffer (DB), a built-in maintenance feature has been incorporated that allows the programmer to write into the DB as a register, thereby allowing a succession of data words to be sequenced through the DB and read back via a read from the DB. In this way, the data path and parity generator and checker for the Massbus can be easily verified.

### 3.3.1 Data Buffer Maintenance Operation

Figure 3-4 shows the data path used when the programmer attempts to read or write the DB register during maintenance operations. The RA, RB, RC, RD and RE - OBUF registers contain associated flags (RA FULL, RB FULL, RC FULL, RD FULL, RE FULL, - OBUF FULL) to designate the status of their respective registers. The RA or RB FULL flags are presented to the programmer via the IR (Input Ready) status bit (bit 06) in the CS2 Control and Status 2 reg ister. When the hardware selected flag (RA FULL or RB FULL) is asserted (register is full), the Input Ready Status is negated, indicating that the register is not available to receive a word. The OBUF FULL flag is presented to the programmer via the OR (Output Ready) status bit (bit 07) of the CS2 Control and Status 2 register. When the OBUF FULL flag is asserted, it indicates that the contents of OBUF are full (has a valid data word) and that the programmer can read the data buffer.

The data path also contains three steering signals used to alternate the flow of data between the right- and left-hand sides of the data path: RB ENA, RD ENA, and MXR SEL. When the system is first initialized, by asserting INIT on the Unibus, or by setting the CLR bit in the CS2 register, the RB ENA flip-flop is pointing to the RA register, the RD ENA flip-flop is pointing to the RC register, and the MXR SEL logic is pointing to the RC register. This primes the right-hand side of the data path to accept the first data word.

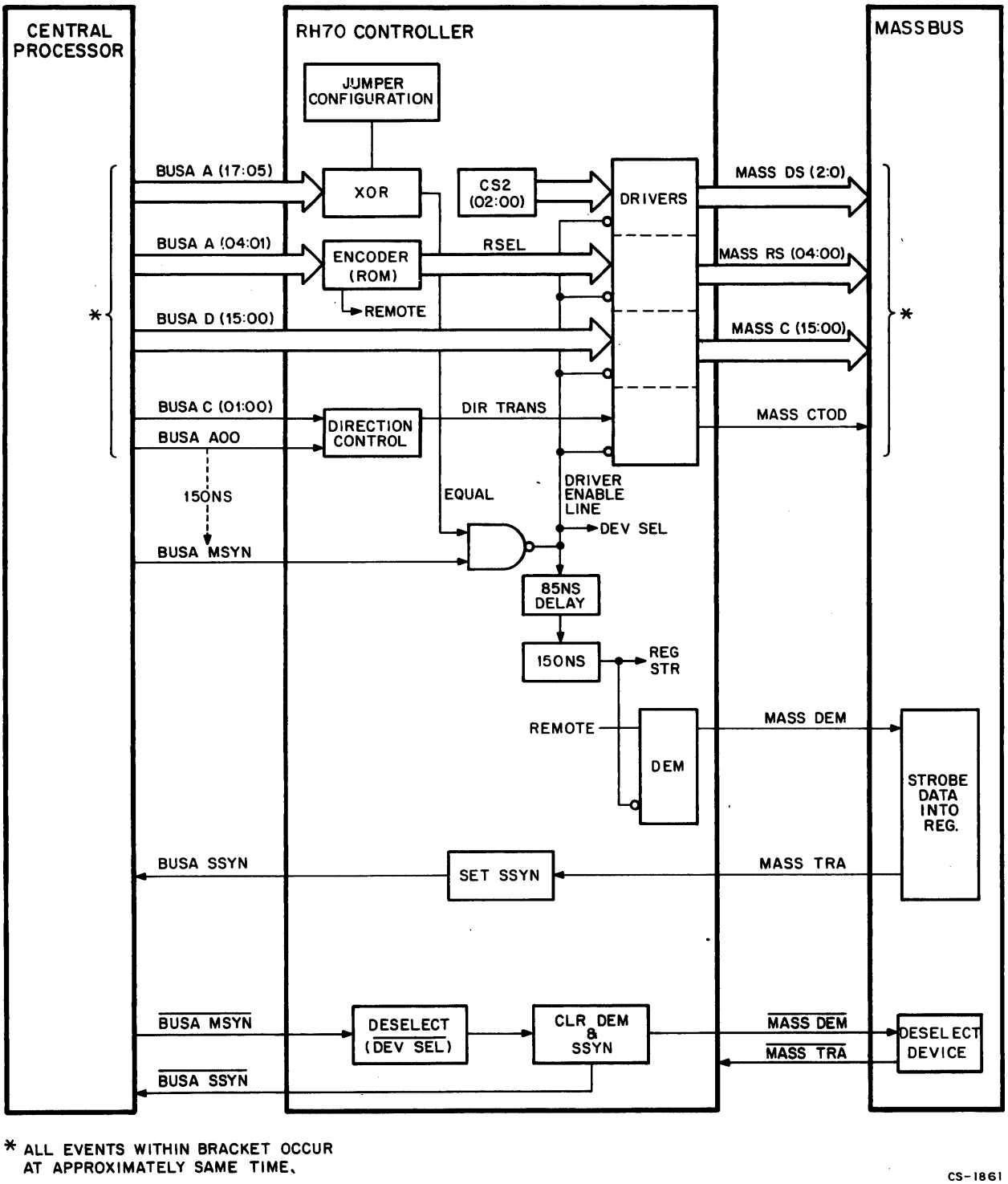



Figure 3-3 Reading Remote Register Interface

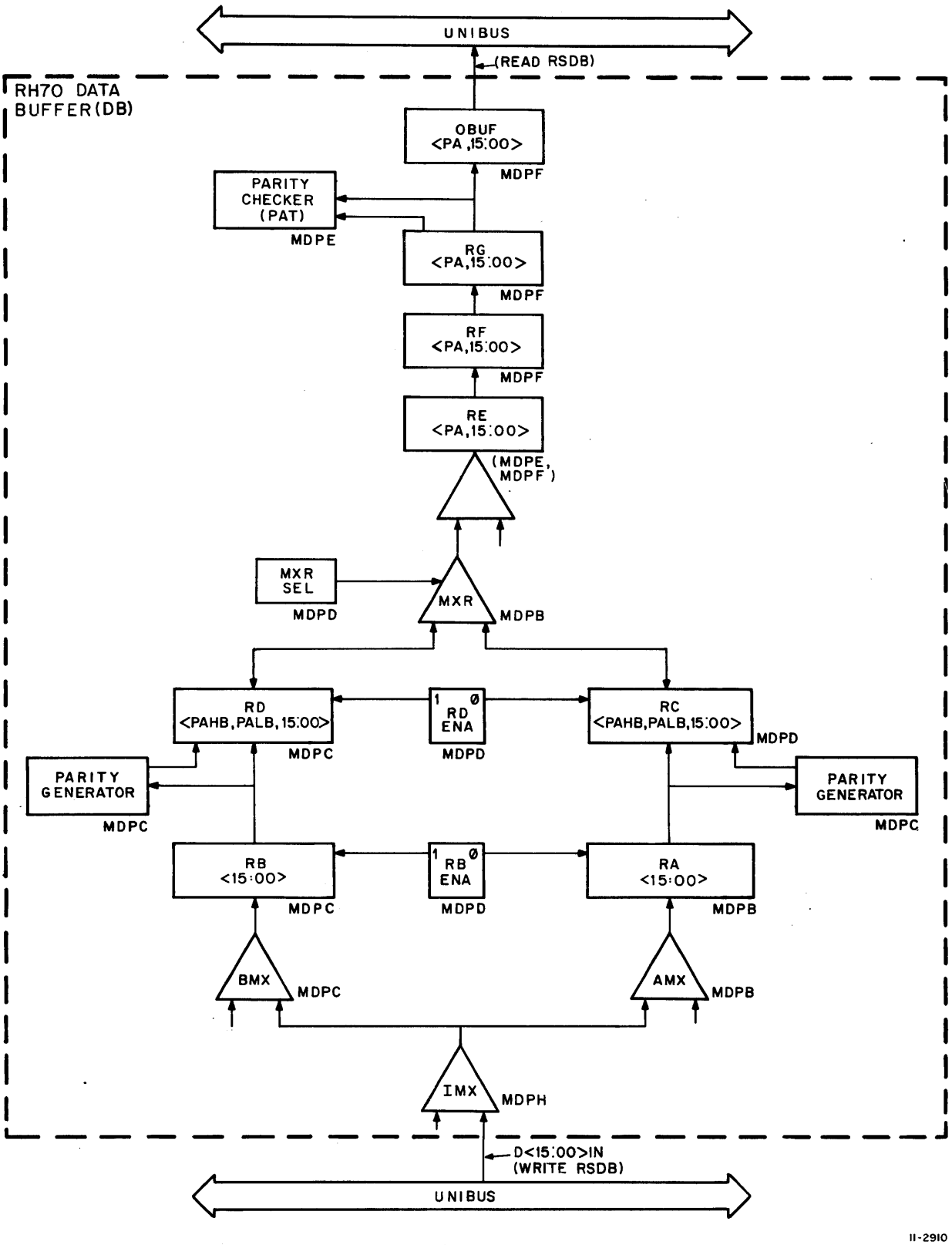



Figure 3-4 Data Buffer Maintenance Operation

**3.3.1.1 Writing the Data Buffer During Maintenance Operation** – To understand maintenance operation of the data buffer, the following description assumes that eight words are to be sequentially loaded into the data buffer and none of the words are to be read out.

*WORD 1* – The first word loaded in the DB by the programmer is transferred from the Unibus to IMX, AMX and then to the RA register. This action causes two events: the RA FULL flag is asserted (which negates the IR Status in CS2) and the RB ENA flip-flop is toggled to point to the RB register (which, if empty, will cause the IR Status to again be asserted). Since the RC register is empty (RC Full flag negated) and the RA register is full, the data word in RA sequences to the RC register. This causes the RC FULL flag to be asserted, the RA FULL flag to be negated, and the RD ENA flip-flop to toggle to the RD register. The data word then sequences to the RE register through the MXR since the MXR SEL was pointing to RC. The RC FULL flag is cleared, the RE FULL flag is asserted, and the MXR SEL switches to the RD register. Now, the logic is set up to enable the left-hand side of the data path. The data word in RE is transferred to RF, to RG, and then to OBUF, which causes the OBUF FULL flag to be asserted.

*WORD 2* – The second word loaded into the DB by the programmer is transferred from the Unibus to IMX, BMX, and then to the RB register. The RB FULL flag is asserted and the IR (Input Ready) status is cleared. The RB ENA flip-flop is toggled to point to RA again (which causes IR to be asserted, since RA is empty). Since the RD register is empty (RD FULL flag negated) and the RB register is full, data word 2 sequences to the RD register. This negates the RB FULL flag, asserts the RD FULL flag and toggles the RD ENA flip-flop to point to the RC register again. The word in RD is then transferred to RE via the MXR SEL logic, since the MXR SEL is pointing to RD. At this point, the MXR SEL is changed to point to RC again. Now, the right-hand of the data path is primed to receive the next data word. Data word 2 in RE is transferred to RF, then to RG, and remains there since data word 1 is still in OBUF.

*WORD 3* – As previously stated, the right-hand side of the data path is primed and the third word loaded by the programmer is transferred from the Unibus to IMX, AMX, RA, RC through the MXR to RE, and finally to RF. The description of how this word sequences through the DB is similar to that described for word 1. The difference is that word 3 remains in RF since word 2 is in RG and word 1 is in OBUF. All the pointers are now primed to enable the left-hand side of the data path.

*WORD 4* – The fourth word is transferred from the Unibus to IMX, BMX, RB, RD, and finally to RE. The description as to how this word sequences through the DB is similar to that described for data word 2; however, the fourth data word remains in RE, since word 3 is in RF, word 2 is in RG, and word 1 is in OBUF. The pointers are now all primed to enable the right-hand side of the data path.

*WORD 5* – The fifth word loaded by the programmer is transferred from the Unibus to IMX, AMX, RA and then to the RC. The word can sequence no further because RE is already full with the fourth word. The RB ENA and RD ENA flip-flops are toggled to point to RB and RD, respectively; however, the MXR SEL is not affected since no word was transferred through the mixer in this case. The left-hand side of the data path is now primed to accept the sixth data word.

*WORD 6* – The sixth word loaded by the programmer is transferred from the Unibus to IMX, BMX, RB and then to RD. Both the RB ENA and RD ENA flip-flops are toggled to point to RA and RC, respectively, in order to prime the right-hand side of the data path to accept data word 7. The MXR SEL again remains unchanged, since no word is sequenced through the mixer. At this point, word 1 is in OBUF, word 2 is in RG, word 3 is in RF, word 4 is in RE, word 5 is in RC and word 6 is in RD. Since both RC and RD are full, the RD ENA flip-flop can now be ignored, as is the case with the MXR SEL.

*WORD 7* – The seventh word loaded into the DB is transferred from the Unibus to IMX, AMX, and then to RA. The word remains in RA since word 5 is still in RC. The RB ENA flip-flop is toggled to prime the left-hand side of the data path in order to accept the eighth data word.

*WORD 8* – The eighth word loaded is transferred from the Unibus to IMX, BMX, and then to RB. The eighth word remains in RB since all the remaining registers in the DB are filled. The RB ENA flip-flop is toggled to point to RA. Since a word is in RA (RA Full flag asserted), the IR (Input Ready) status is negated.

The programmer can verify that the DB is full by examining the OR (Output Ready–bit 07 of CS2) and IR (Input Ready–bit 06 of CS2) status bits. If OR is asserted and IR is negated, it indicates that a word is in OBUF and words are in RA and RB, and due to the "bubble" technique of word sequencing, it is evident that the data buffer is full. If the programmer attempts to write a ninth word into the DB, he would overwrite the entire contents of RA and would cause the RB ENA flip-flop to toggle to point to RB.

**3.3.1.2 Reading the Data Buffer During Maintenance Operation** – When the programmer does a read from the DB register, the contents of OBUF is supplied to the Unibus as data and the OBUF FULL flag is cleared (OR status bit negated). When this occurs, the contents of RG (word 7) is sequenced to OBUF (OBUF FULL sets and OR if asserted), RF sequences to RG, RE to RF, RC to RE (since the MXR SEL was pointing to RC) and RA to RC. This action causes the MXR SEL to now point to RD, the RD ENA flip-flop to point to RD and the RB ENA flip-flop to remain unchanged (pointing to RA). At this point, all the registers in the Data Buffer are filled, except for the RA register, and the IR status indicator is asserted in the CS2 register to indicate that space is available for additional data words.

When the second word is read from OBUF, the OBUF FULL flag is cleared. The contents of RG (word 6) is sequenced to OBUF, RF to RG, RE to RF, RD to RE (since the MXR SEL is pointing to RD), and RB to RD. This process causes the MXR SEL to switch to RC, the RD ENA flip-flop to point to RC and the RB ENA flip-flop to re main unchanged (pointing to RA). At this point, word 3 is in OBUF, word 4 is in RG, word 5 is in RF, word 6 is in RE, word 7 is in RC and word 8 is in RD, with both RA and RB cleared. As successive reads are performed, the words are sequenced out of the DB in the same order in which they are loaded into DB.

**3.3.1.3 Parity Generation/Checking During Maintenance Operation** – Whenever a word is clocked into RA and then to RC, the parity checker/generator circuit (shown between RA and RC) generates odd parity for each byte as the word is sequenced to RC. For example, assume that the low byte contains all 1s; in this case, a parity bit is generated. Assume that the high byte contains all 0s; in this case, a parity bit is also generated. A similar parity generator/checker is located between RB and RD, and as a word is transferred from RB to RD, odd parity for each byte is created.

As the data word sequences the RE, the parity bits are X-ORed, resulting in one parity bit for the word. This parity bit is transferred along with the data word until the data word reaches OBUF. There is a third parity generator/checker prior to OBUF. This circuit checks the parity of the data word to ensure that it is odd parity. If it is not, the parity checker will cause the MDPE (Massbus Data Parity Error) condition (bit 8 of CS2) to be asserted, indicating that a hardware failure occurred between inputting the data word into the data buffer and the outputting of the data word from the data buffer.

Additional flexibility is provided the programmer for maintenance checking of these parity circuits by allowing bad (even) parity to be *generated* as words are input or by allowing bad (even) parity to be *checked* as the words are input. The IPCK bits control the parity generators between RA and RC and between RB and RD. The PAT bit controls the parity checker between RG and OBUF.

### 3.3.2 Write Commands (Data Buffer Operation)

In a write operation, data is transferred from memory to the RH70 Massbus controller, to the Massbus, and subsequently to the drive. The data buffer in the RH70 is the device that accepts data to be written on or read from the drive (Figure 3-5). In a write operation, words are obtained from memory in one of four ways: single word, double word, words whose addresses are sequentially incremented and words whose addresses are sequentially decremented. When one of these operations is specified, the RH70 data buffer is initialized for that type of operation. When bus address bit 01 (BA 01) is a 0 (even word address), the right-hand data path (RA and RC) is initialized; when this bit is a 1 (odd word address), the left-hand data path (RB and RD) is initialized.

For a data transfer, the program specifies the bus address, bus address extension, word count, desired address and the data transfer command code – a write command in this case. When the write command is loaded and the data buffer is full, or word count overflow or an error condition occurs, the RUN line is asserted on the Massbus and the drive starts searching for the sector specified by the desired address. When the desired address is found, the drive sends SCLK (Sync Clock) signals to the RH70. The RH70 returns SCLK as a WCLK (Write Clock) signal, along with the data word. On the trailing edge of WCLK (SCLK plus the cable delay), the drive strobes the data word into its data buffer. This process continues until either word count overflow is detected or until an error condition is asserted.

The normal operation of the data buffer is to transfer double words as the memory addresses are sequentially incremented. The hardware also implements double-word transfers as the memory addresses are sequentially decremented. In certain conditions, it is necessary to do single-word transfers. These conditions are enumerated below.

1. Word Count – If the word count indicates that the word to be transferred is the last word, a single word transfer will occur.

2. Bus Address Increment Inhibit – If this bit (BA1–bit 3 of CS2) is asserted, the memory addresses are not incremented and the same data word is written in successive locations in the drive.

3. If the memory address of the data word is on a single-word boundary, it is necessary to do a single-word transfer to arrive on a double-word boundary. When this occurs, normal double-word transfers can be accomplished. If memory addresses are being incremented, and the least significant digit of the address is a 0 or 4, the word is on a double-word boundary. In this case, Bus Address bit 01 (BA 01) is a 0. If the least significant digit of the address is a 2 or 6, the word is on a single-word boundary and BA 01 is correspondingly a 1 (Figure 3-6).

   The reverse case is true when memory addresses are being decremented. In this instance, a least significant digit of 2 or 6 specifies a double-word boundary (BA 01 equals a 1) and a least significant digit of 0 or 4 specifies a single-word boundary (BA 01 equals a 0).

The following paragraphs describe a write data transfer using incremented memory addresses followed by a write data transfer using decremented memory addresses.

**3.3.2.1 Incremented Memory Addresses** – To understand the data buffer, assume that BA 01 is a 0, which places the memory address on a double-word boundary and initializes the right-hand data path with the RB ENA pointer pointing to RA. It is further assumed that a double-word data transfer is to be performed, followed by a single-word transfer (i.e., a word count of 3).

The low word from memory is applied to RA via AMX and the high word from memory is simultaneously applied to RB via BMX. During double-word operation, the RB ENA pointer is overridden. This can be thought of as adding two to the RB ENA pointer which is actually a flip-flop; adding two to this device reverts it to its original state. Consequently, after this double-word transfer, the RB ENA pointer will remain pointing to RA. In addition, the RD ENA pointer is not employed during a write command and can be ignored.

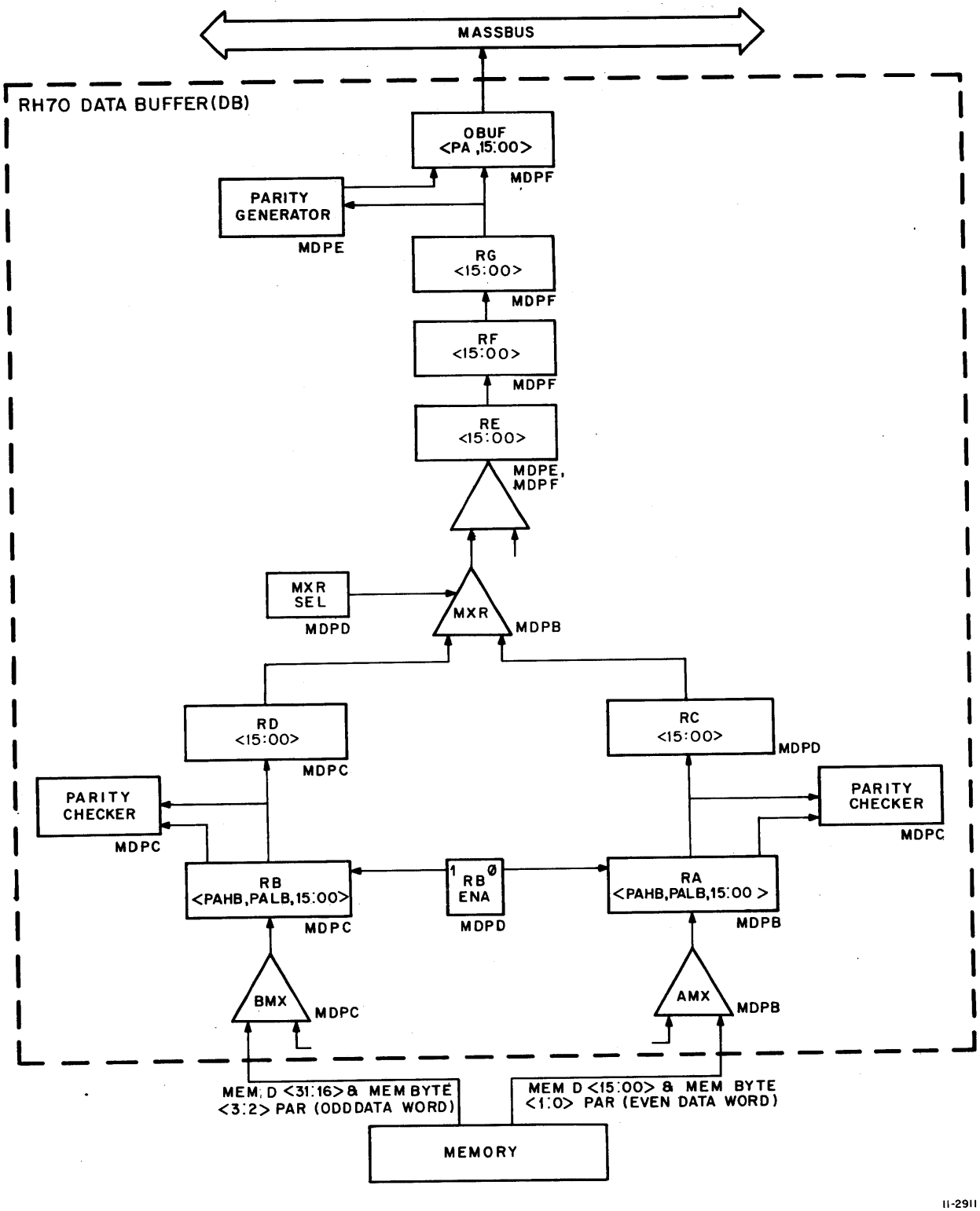



Figure 3-5 Write Command Data Path

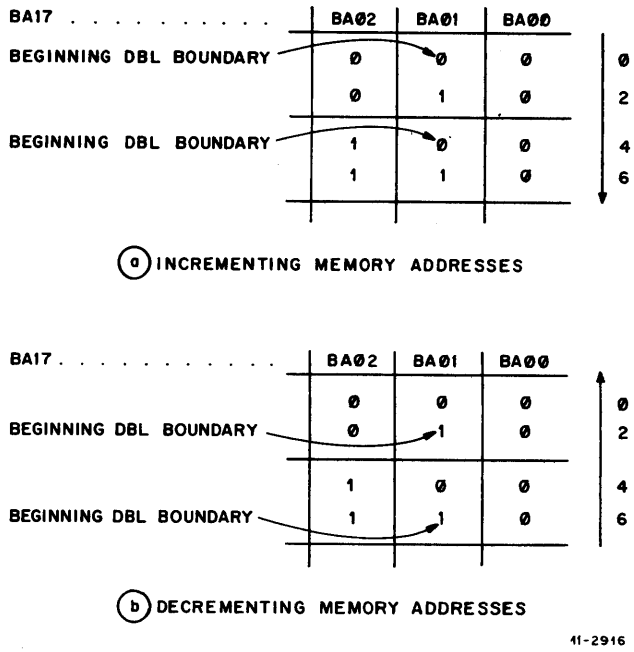



Figure 3-6 Data Word Boundaries

When the double word from memory is clocked into RA and RB, the RA FULL and RB FULL flags are asserted. Since RC and RD are empty, the double word is clocked into RC and RD, respectively, clearing the RA and RB FULL flags and asserting the RC FULL and RD FULL flags. Each data word consists of two bytes, plus a parity bit for each byte. The byte parity is checked by a parity checker as the double word is transferred from RA to RC and from RB to RD. If the parity is correct, the act of transferring a word from RA to RC or a word from RB to RD will initiate a new memory cycle. If the parity is incorrect, the data parity error is flagged and the memory transfer is frozen in order to allow the programmer to determine the address of the incorrect data word.

NOTE

**The data word may or may not be written in the drive. This depends on whether the parity error is synchronized with the SCLK from a previous word, or with the SCLK associated with the word in question. In the former case, the data word will remain in the data buffer; in the latter case, the word will be written in the drive.**

Since RC and RD are full, and the MXR SEL is pointing to RC, the word in RC is transferred to RE, clearing the RC FULL flag, and switching the MXR SEL to RD. The word in RE sequences through the data buffer and is followed by the word in RD which is now transferred to RE. As a result, the RD FULL flag is cleared and the MXR SEL points once again to RC. The double word has thus been converted into two single words - the low word followed by the high word. As the data word is transferred from RG to OBUF, a parity bit is generated if required (for odd parity). This ensures correct parity for the data as it is clocked onto the Massbus.

For descriptive purposes, assume that the programmer had requested a three-word transfer and a single word must follow behind the double word which is now located in OBUF and RG. This word from memory is applied to RA via AMX, since RB ENA is pointing to RA and it is an even word. Since RC is empty, the word is transferred to RC, the RC FULL flag is asserted, the RA FULL flag is cleared, and the RB ENA pointer is switched to RB. Since the MXR SEL is pointing to RC, the word in RC is transferred to RE via the mixer. The MXR SEL is switched to RD, the RC FULL flag is cleared, and the single data word continues to sequence through the data buffer to RF. Two memory cycles have been performed to read three words from memory. The first word is in OBUF, the second in RG and the third in RF, ready to be transferred to the drive via the Massbus.

**3.3.2.2 Decremented Memory Addresses** - To understand the operation using decremented memory addresses, assume that BA 01 is set to a 0, indicating that the data word is on a single-word boundary. In order to perform double-word memory cycles, the memory address must be on a double-word boundary. To accomplish this, a single word transfer is first performed; then the next memory address will be on a double word boundary.

Since BA01 is a 0, the right-hand data path (RA and RC) is initialized. The single word from memory is clocked into RA via AMX. The RB ENA pointer is switched to RB and the RA FULL flag is asserted. Since RC is empty, the word in RA is transferred to RC, the RA FULL flag is cleared, and the RC FULL flag is asserted. As the word is transferred from RA to RC, a parity check is performed. If parity is correct and if RA and RB are empty, a new double-word memory cycle is initiated. If parity is incorrect, the memory transfer is frozen as previously described.

Since the MXR SEL is pointing to RC and since RE is empty, the word in RC is transferred to RE. The RC FULL flag is cleared and the MXR SEL is switched to RD. The word in RE continues to sequence through the data buffer and as the word is transferred from RG to OBUF, a parity bit is generated, if required.

If RA and RB are empty and no parity error is detected, the transfer of the word from RA to RC initiates a double-word memory cycle (BA01 is a 1 which now points to a double-word boundary). The double word is clocked into RA and RB, then into RC and RD, as previously described. Since the RB ENA pointer does not switch for double words and the RD ENA pointer is not used for write operations, they can be ignored. The MXR SEL is pointing to RD and the high word in RD is transferred to RE, followed by the low word in RC. The description of the words sequencing through the data buffer is similar to that described for incremented memory addresses.

The important distinction is that the low word follows the high word of a double-word block in decremented memory addresses and the MXR SEL points to RD for a double-word boundary, which is the reverse of what occurs in incremented memory addressing.

### 3.3.3 Read Command (Data Buffer Operation)

In a read data transfer, data is transferred from the drive to the Massbus, to the data buffer in the RH70 Massbus Controller and subsequently to memory. The data buffer in the RH70 is the device that accepts data to be written on or read from the drive (Figure 3-7). In a read operation, single words are clocked into the data buffer, via the Massbus, and are normally assembled into double words for subsequent transfer to memory.

For a data transfer, the program specifies the bus address, bus address extension, word count, and the data transfer command code – a read command in this case. When the read command is loaded, the RUN line is asserted on the Massbus. When the Massbus device is ready to begin transferring data, it asserts the data on the Massbus and sends it to the RH70, accompanied by an SCLK signal. On the trailing edge of SCLK, the data is strobed into the RE register in the data buffer. The data sequences from RE to RF, to RG and the OBUF. Parity is checked as the data is transferred from RG to OBUF.

NOTE

**Parity is calculated on the 16 data bits, the parity bit, and the exclusive-OR of data bits 16 and 17 from the Massbus. Bits 16 and 17 are written as 0s during write commands. If a parity error occurs, a Massbus Data Parity Error (MDPE) is flagged.**

The data word in OBUF is transferred to RA (if BA01 = 0) or is transferred to RB (if BA01 = 1). If double-word operation is occurring, the next data word is assembled with this one as a double-word in RA and RB for transfer to RC and RD and subsequently to memory. As previously described, during incremented memory operations, addresses whose least significant bit is a 0 or 4 are on a double-word boundary while addresses whose least significant bit is a 2 or 6 are on a single-word boundary. The reverse holds true for decremented memory addressing.

**3.3.3.1 Incremented Memory Addresses During Read Operation** – To understand double-word transfers during a read operation, assume that BA 01 = 0 so that the RB ENA pointer points to RA and the RD ENA pointer points to RC. The word reaching OBUF from the drive is clocked into the RA register. At this point, the RA FULL flag is asserted and the RB ENA pointer switches to RB. Since this is a double-word transfer, the data path logic prevents the word in RA from being clocked into RC until a word is loaded in RB. Now, a double word is assembled in RA and RB and, since RC and RD are empty, the double word is clocked into RC and RD. The RC FULL and RD FULL flags are asserted. The RA FULL and RB FULL flags are negated, and the RD ENA pointer is incremented twice, which effectively leaves it pointing to RA. The act of transferring the word from RA and RB to RC and RD initiates a memory request.

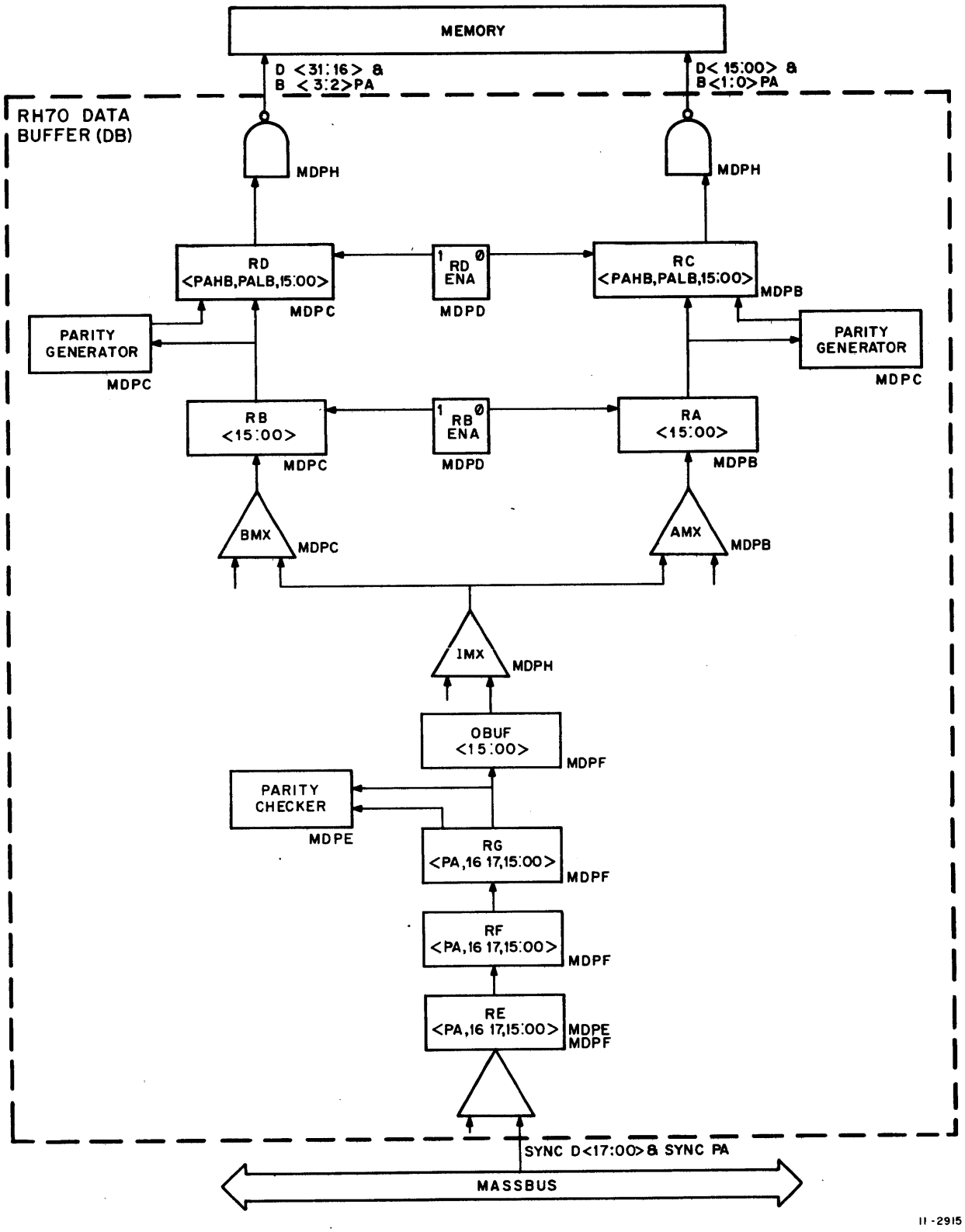



Figure 3-7 Read Command Data Path

In single-word transfers, the RD ENA pointer ensures that the RC or RD register is clocked at the proper time. After memory acknowledges the request and completes the transfer, the RC FULL and RD FULL flags are cleared. The process continues with data being clocked into RE, RF, RG, OBUF, and subsequently to RA and RB, where it is assembled into double-words for transfer to RC and RD.

NOTE

**The double-word operation will normally take place if the word is on a double-word boundary, if it is not the last word, or if the BAI (Bus Address Increment Inhibit) bit is negated.**

Parity generators attached to RA and RB examine each data byte and generate an odd parity bit, if required. Each 16-bit word consists of 2 8-bit bytes and an associated parity bit for each byte. Both data and parity are sent to memory.

**3.3.3.2 Decremented Memory Addresses During Read Operation** – To understand read data transfer operations using decremented memory addresses, assume that BA 01 is a 0, which primes the right-hand data path and indicates that the data word is on a single-word boundary.

The drive places a data word on the Massbus with a SCLK signal. On the trailing edge of SCLK, the RH70 strobes the word into RE and it sequences through RF, RG, OBUF and into RA, since the right-hand data path is enabled. The RA FULL flag is asserted and the RB ENA pointer is switched to RD. The act of transferring the contents of RA to RC initiates a memory request and the data word is then transferred to memory. At the completion of the memory transfer, the RC FULL flag is cleared.

The next data word is then on a double-word boundary and is transferred to the RE register from the drive via the Massbus. The word sequences from RE to RF, RG, OBUF and then to RB, since the left-hand data path is enabled. The RB FULL flag is asserted and the RB ENA pointer is switched to RA. However, since a double-word transfer is specified, the contents of RB will not be transferred to RD until both RC and RD are empty and both RA and RB are full. The next data word from the drive sequences through the data buffer and is clocked into RA since the right-hand data path is primed. Now, a double word is in RA and RB, RC and RD are empty and the double word is transferred into these locations. This clears the RA FULL and RB FULL flags, asserts the RC FULL and RD FULL flags, and initiates a memory request, causing the double word to be transferred to memory. Successive double words continue to be transferred in this manner.

### 3.3.4 Write Check Command (Data Buffer Operation)

In the write-check operation, data previously written onto the drive is compared with the corresponding data from memory. Using this technique, transmission errors from memory to the drive can be detected.

The data words from the drive are transferred to the RE register in the data buffer via the synchronous portion of the Massbus (Figure 3-8). The data sequences from RE to RF, RG, and then to OBUF. This portion of the data path is similar to that described for the Read command (Figure 3-7).

Data from memory is normally input as double words to AMX and BMX and then to RA and RB, respectively. In the case of a single-word transfer, the data word is clocked into RA (if BA01 = 0) or into RB (if BA01 = 1). This portion of the data path is similar to that described for a write operation (Figure 3-5).

NOTE

**As previously described in the write operation, a double word is clocked into RA and RB. This causes the RA FULL and RB FULL flags to be asserted. The RB ENA pointer is overridden for double-word transfers. If both RC and RD are now empty, the double word is clocked into RC and RD, the RC FULL and RD FULL flags are asserted and the RA FULL and RB FULL flags are cleared. A parity check is performed on the data as it is clocked into RC and RD. If a parity error is detected, a data parity error is flagged and the next memory request is inhibited. The act of clocking the double word into RC and RD with no parity error initiates another memory request.**

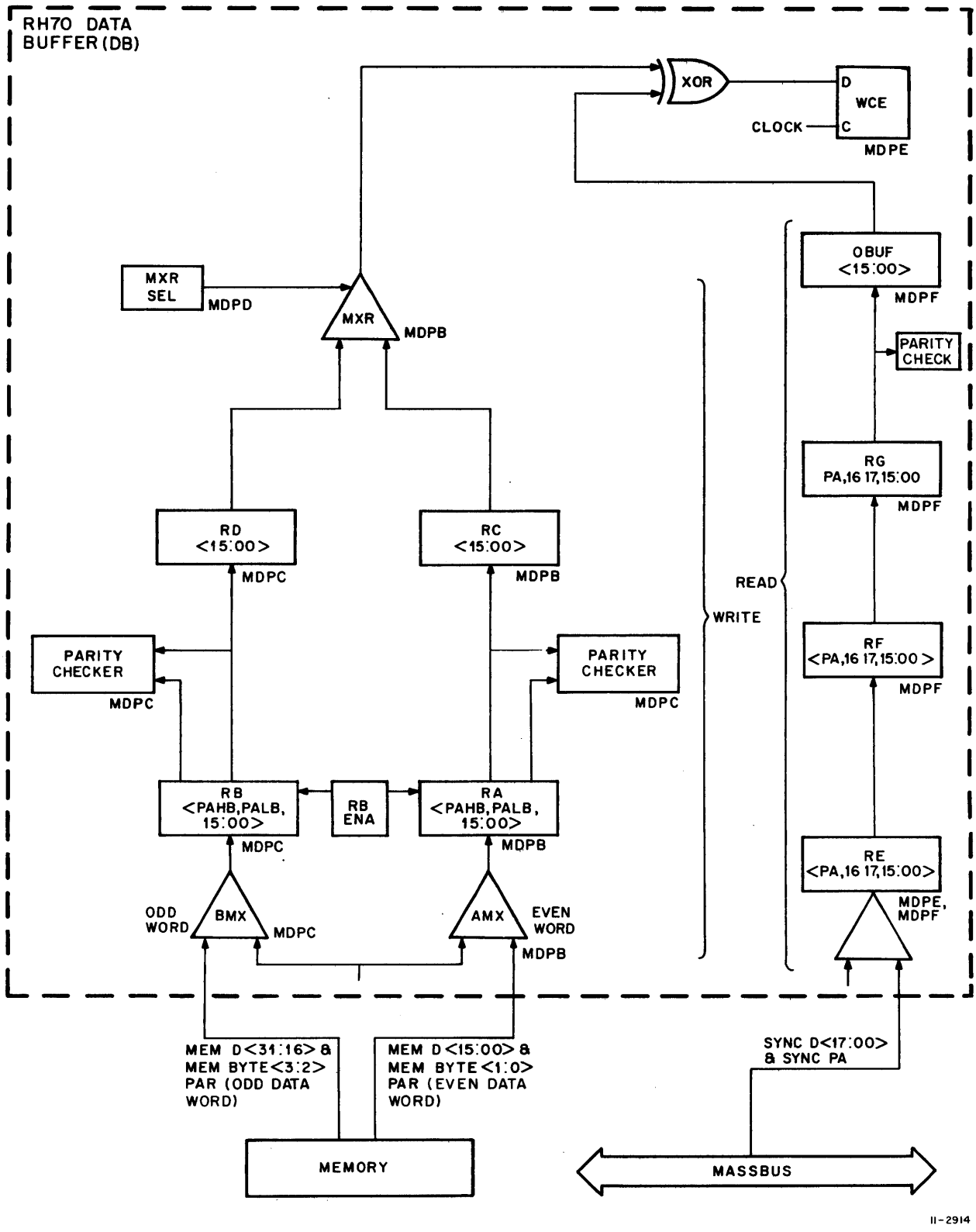



Figure 3-8 Write-Check Command Data Path

To understand the write-check operation, assume that a data word from the drive has been transferred to the RE register and has sequenced up to OBUF; also assume that a double word from memory has sequenced into RC and RD. Also, assume that the MXR SEL is pointing to RC. In this case, the output of RC is applied to one input of the exclusive-OR (XOR) network and the output of OBUF is applied to the other input of the XOR network. The RC output is the data word from memory and the OBUF output is the corresponding word from memory which was previously written onto the drive. These two words are compared in the XOR network on a bit-by-bit basis. If any of the bits do not compare, the XOR network yields a 1 which is used to set the WCE (Write-Check-Error) flip—flop when it is clocked. If the data words do compare, the WCE flip-flop remains in the reset state.

When the WCE flip-flop is clocked by the clock signal, the OBUF FULL flag is cleared (if no WCE occurs) and the RC or RD flip-flop is cleared (whichever was being selected by MXR SEL). In this case, it is the RC flip-flop. This action releases the word in RC and OBUF to allow subsequent words to be write-checked. At the same time, the MXR is switched to RD. The data word in RD is now compared with the next data word from the drive which has sequenced up through OBUF. The MXR switches back to RC after this check and the process continues.

For write-checking memory addresses which are decremented, the procedure is similar, except that the high data word is write-checked before the low data word. IF BA01 is a 1, the left-hand data path is primed, which means hat the high word in RD is write-checked before the low word in RC. If BA01 is a 0, the right-hand (low word) data path is primed and, in decrementing operations, this word is on a single-word boundary.

## 3.4 RH70/CACHE INTERFACE

Data Transfers between the RH70 Massbus Controller and Cache take place over the RH70/Cache Interface. The transfers are asynchronous in nature and normally consist of two data words (32 bits, plus four parity bits). The Cache interfaces with up to four RH70 Massbus Controllers and contains the necessary priority arbitration logic to select the appropriate controller when multiple requests are initiated.

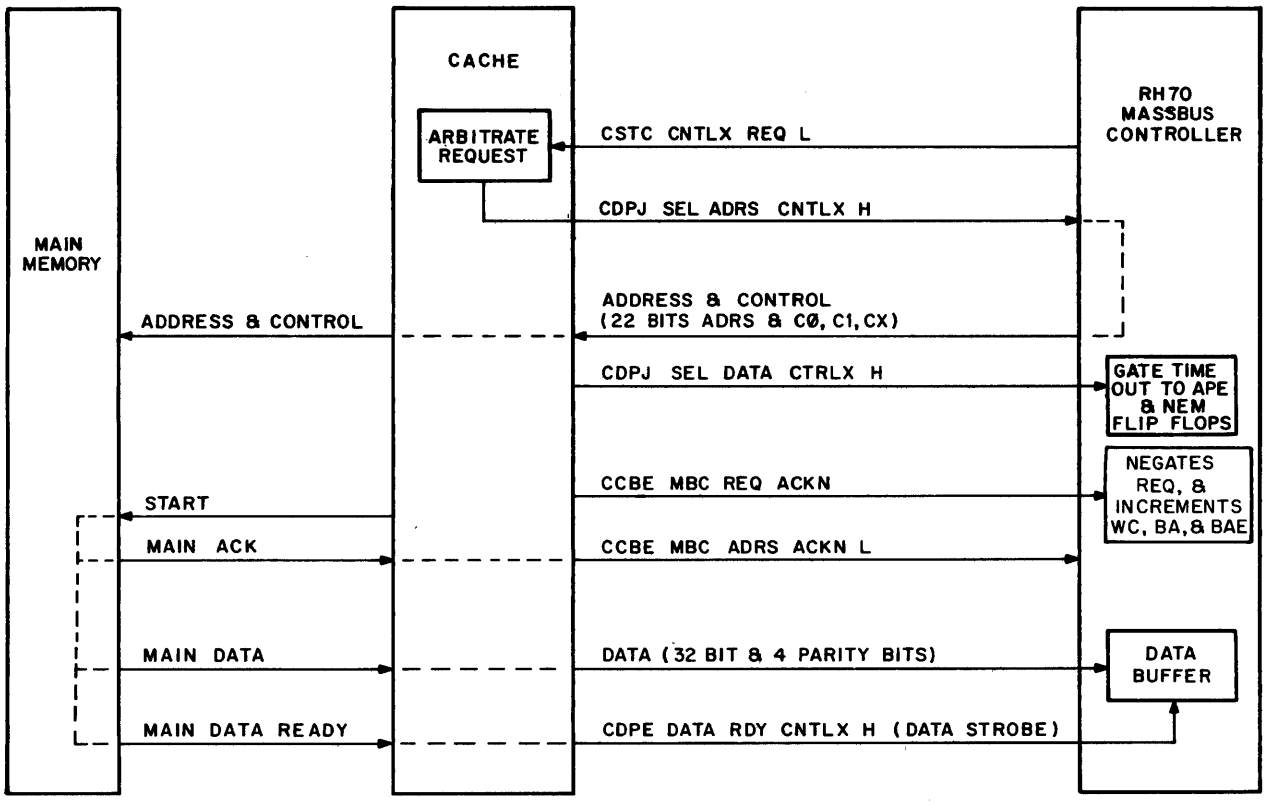
The backplane slots are designated A, B, C and D for the various controllers which it can accommodate. The logic for the controllers is identical; therefore, in order to make one set of logic diagrams common to all controllers, certain signals are designated with an X in the signal designation. Examples of this are CSTC CNTLX REQ L, CDPJ SELADRS CNTLX H and CDPJ SELDATA CTRLX H. The X in these signal names designates A, B, C or D, depending upon which controller is specified. For example, the signal CSTC CNTLX REQ L for Controller A is CSTC CNTLA REQ L. The following paragraphs provide the interface description of write data transfers (data read from memory to be written on a Massbus device) read data transfers (data read from a Massbus device to be written into memory) and write-check data transfers (data read from memory to be compared with data from a Massbus device in the RH70).

### 3.4.1 Write or Write-Check Operation (Read from Memory)

Figure 3-9 is an interface diagram showing the relationships among the RH70/Cache Interface and the Cache/Memory Bus Interface. Figure 3-10 is a timing diagram showing the RH70/Cache timing relationships.

When the RH70 wishes to make a memory access, it issues a CSTC CNTLX REQ signal. For example, if Controller A is making the request, CSTC CNTLA REQ is issued. The Cache receives the RH70 request along with requests from other devices and arbitrates them to determine which controller will access memory. The Cache then transmits CDPJ SELADRS CNTLX H to the Controller which has been selected. If Controller A was selected, Cache issues CDPJ SELADRS CNTLA H.

Upon receipt of this signal, the RH70 gates the address and control bits (C0, C1, and CX) to Cache. The Cache latches the address and control bits for subsequent transfer to Main Memory.




NOTE: This figure assumes no Address Parity Error and no Non-Existent memory, both of which cause CCBD MBC TIMEOUT. The RH70 monitors for these conditions while CDPJ SEL DATA CNTLX H is asserted.

Figure 3-9 Memory Interface Write or Write- Check Operation (Read from Memory)

The Cache then transmits CDPJ SELDATA CTRLA H to the RH70. During write or write-check operations, this signal is used to gate CCBD MBC TIMEOUT from Cache. CCBD MBC TIMEOUT will occur only as a result of addressing a nonexistent memory or as a result of an address parity error on the Main Memory Bus. Either condition causes an error status to be indicated in the RH70 (APE-bit 15 of CS3 and NEM-bit 11 of CS2). The Cache then issues a CCBE MBC REQ ACKN signal to the RH70 that it can now unassert the request signal and the address and control bits.

After the Cache issues CCBE MBC REQ ACKN, it issues a START signal to Main Memory. This causes the address and control bits, which were latched in Cache, to be transferred to Main Memory. A memory cycle is initiated and memory issues MAIN ACK, indicating that memory is responding properly. The assertion of MAIN ACK inhibits the TIMEOUT signal (which denotes an error condition when asserted). After receiving MAIN ACK from memory, the Cache issues CCBE MBC ADRS ACKN L to the RH70. In the RH70, MBC ADRS ACKN is made into a 50-ns pulse and causes CSTC READ DONE L. The trailing edge of READ DONE clears the memory cycle control logic in the RH70.

NOTE

**If ADDR ACK is not received within 10 microseconds after START, Cache issues a CCBD MBC TIMEOUT signal and the RH70 examines the Address Parity Error line. If TIMEOUT and Address Parity Error are asserted, the CSTC APE flip-flop is set. If TIMEOUT is asserted and there is no address parity error, the CSTC NEM (Non-Existent Memory) flip-flop is set.**

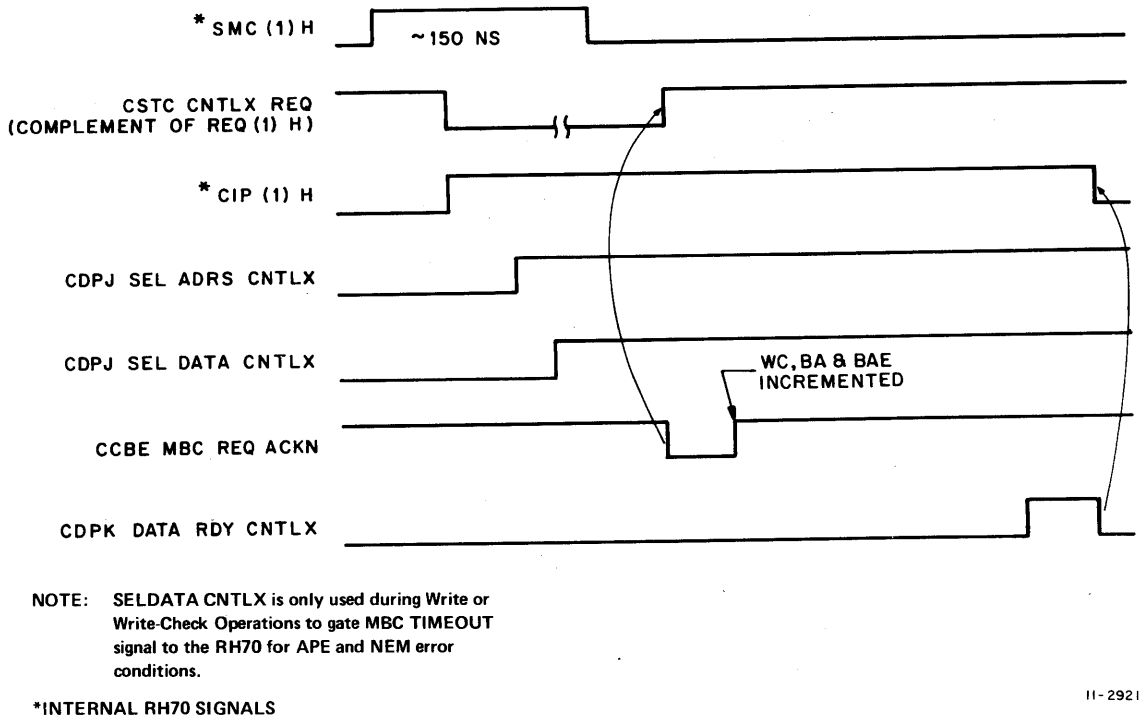



Figure 3-10 Write or Write-Check Command Memory Bus Timing (Read from Memory)

Main Memory then places two words on the data lines of the Main Memory Bus. After a data deskew delay, memory issues MAIN DATA READY. The data words are then routed from Cache to the RH70, selected by the priority arbitration logic (controller A, in this example). The CDPK DATA RDY CNTLA H signal is the MAIN DATA READY signal from Cache to the RH70 and causes the data from memory to be transferred to the data buffer in the RH70. The trailing edge of CDPK DATA RDY CNTLX clears the memory cycle control logic in the RH70. While Cache is transferring data to the RH70, it may already be in the midst of servicing some other request for memory access.

### 3.4.2 Read Operation (Write Into Memory)

Figure 3-11 is an interface diagram showing the relationships among the RH70/Cache Interface and the Cache/Memory Bus Interface. Figure 3-12 is a timing diagram showing the RH70/Cache timing relationships.

When the RH70 wishes to make a memory access, it issues a CSTC CNTLX REQ L signal. The Cache arbitrates this request with other requests in order to determine which device will obtain access to memory. If this RH70 has been selected (assume Controller A), the Cache returns a CDPJ SELADRS CNTLA H signal.

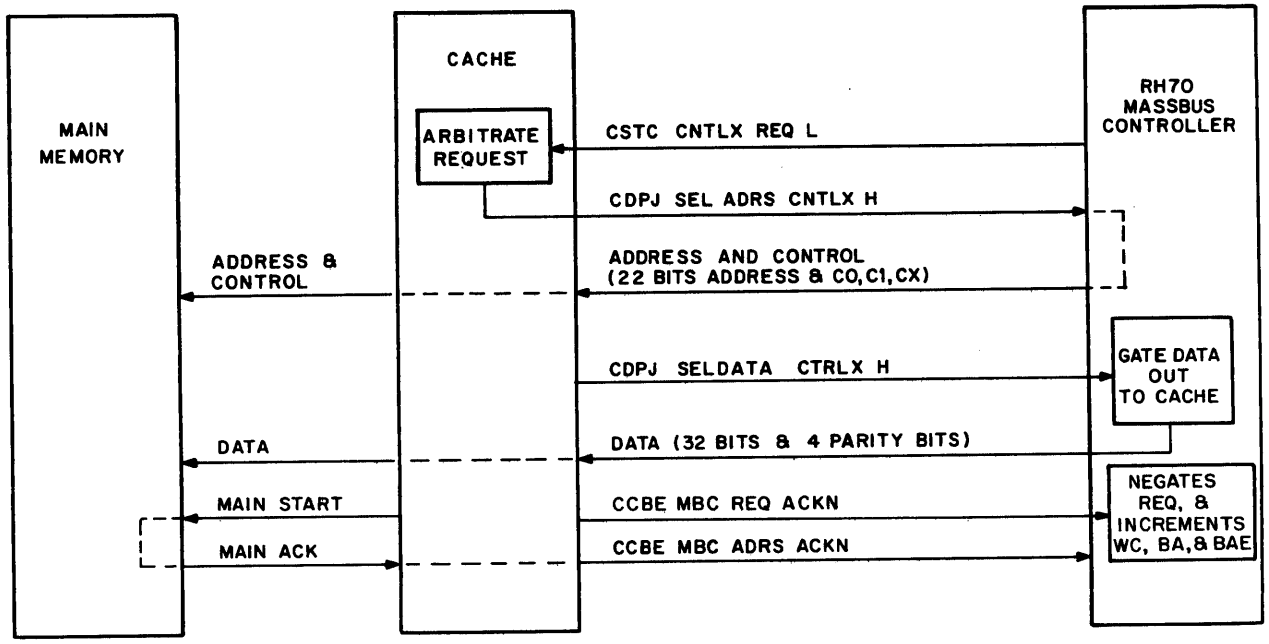
Upon receipt of this signal, the RH70 gates the address and control bits (C0, C1, and CX) to Cache. The Cache latches the address and control bits for subsequent transfer to Main Memory.

The Cache then transmits CDPJ SELDATA CNTLA H to the RH70 (Controller A). This signal allows the RH70 to gate the data from its data buffer onto the RH70/Cache Interface. The data is placed on the Main Memory Bus as long as MAIN OCC is unasserted (indicating that the data lines are free).

**NOTE**

**When a read from memory is being performed, the assertion of MAIN OCC indicates that the data lines are in use.**

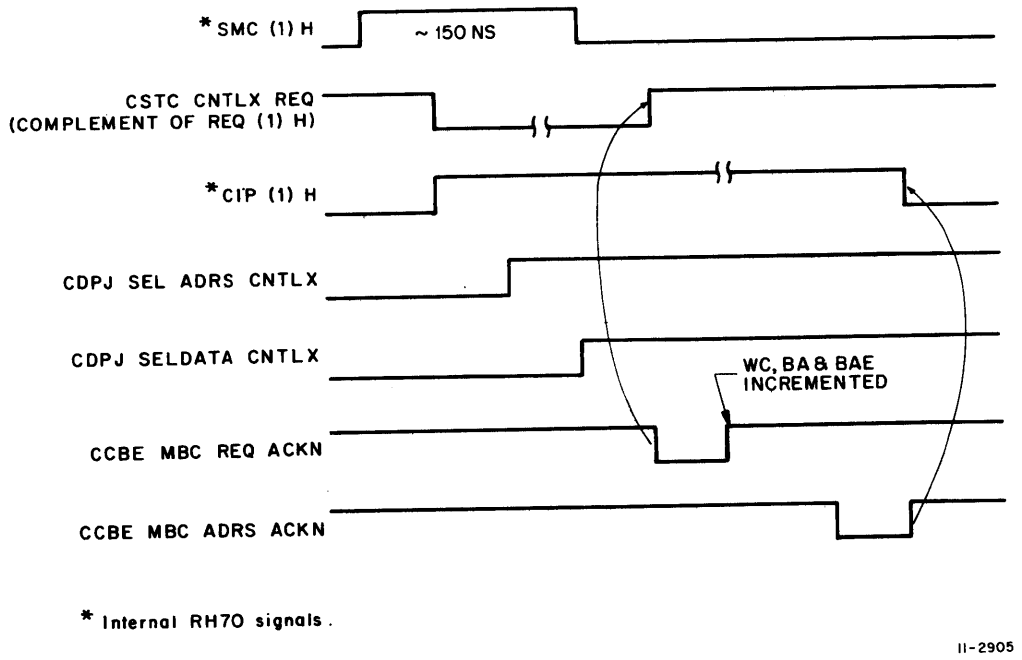
The Cache now issues a CCBE MBC REQ ACKN signal to the RH70 to notify the RH70 that it can negate the request signal and the address and control bits.




NOTE: This figure assumes no Address Parity Error and no Non-Existent memory, both of which cause CCBD MBC TIMEOUT. The RH70 monitors for these conditions while CDPJ SEL DATA CNTLX H is asserted.

II-2904

Figure 3-11 Memory Interface – Read Operation (Write into Memory)




* Internal RH70 signals.

II-2905

Figure 3-12 Read Command Memory Bus Timing (Write into Memory)

After the Cache issues CCBE MBC REQ ACKN, it issues a START signal to Main Memory. This signal causes the address, control bits and data on the Main Memory Bus to be transferred to Main Memory. A memory cycle is then initiated and MAIN ACK is generated by Main Memory and sent to Cache. The memory cycle causes the data to be written into core. Upon receipt of MAIN ACK from memory, Cache issues CCBE MBC ADRS ACKN to the RH70, notifying the RH70 that the Main Memory operation has completed.

The receipt of MAIN ACK by Cache indicates that Main Memory has properly responded, and inhibits the generation of a Main Memory Bus timeout, which would occur if an address parity error occurred or a non-existent memory was addressed. After receiving MAIN ACK from memory, the Cache issues CCBE MBC ADRS ACKN L to the RH70. In the RH70, MBC ADRS ACKN is made into a 50-ns pulse and causes CSTC READ DONE L. The trailing edge of READ DONE clears the memory cycle control logic in the RH70.

NOTE

**This figure assumes that no address parity error has occurred and assumes that non-existent memory has not been addressed. Both of these conditions cause CCBD MBC TIMEOUT. The RH70 monitors for these conditions while CDPJ SELDATA CNTLA H is asserted.**

## 3.5 WRITE COMMAND FLOW DIAGRAM DESCRIPTION

The flow diagram for the write command is shown in Figure 3-13. This paragraph describes the write command, where data is transferred from memory to the RH70 data buffer and is then written onto the drive. The RH70 must be in the Ready State (indicating it is not doing a data transfer command). The assertion of bit 07 (RDY) in the CS1 register puts the RH70 in the Ready state. The program loads the word count, bus address, bus address extension registers and the frame count.

The write command is then loaded, which negates the RDY bit (RH70 becomes busy) and causes the data buffer and control to be initialized.

At this point, the flow divides into two major flows: 1) an RH70/Cache Interface flow, which transfers words from memory to the RH70 data buffer, and 2) a Massbus flow, where words are transferred from the RH70 data buffer to the drive. Both flows operate asynchronously until both DONE and EOS (End of Segment) are asserted (see connector B at right of figure). DONE indicates termination of the RH70/Cache Interface flow and EOS indicates termination of the Massbus flow. When both flows are terminated, RDY is asserted to allow a new command to be loaded. Paragraph 3.5.1 describes the RH70/Cache Interface flow; Paragraph 3.5.2 describes the Massbus flow.

### 3.5.1 RH70/Cache Interface Flow

The RH70 requests that a memory cycle be asserted CNTLX REQ to Cache. At the same time, the RH70 sets a Cycle in Progress (CIP) flip-flop, internally indicating initiation of a memory cycle. The CNTLX REQ signal increments a Start counter. When four such requests have been made (indicating that the data buffer is probably full), the RH70 asserts a START signal. If word count overflow occurs, or if an error condition is posted, the START signal is asserted previous to this. This signal primes the Massbus cycle (Paragraph 3.5.2).

At the same time this is happening, the RH70 is waiting for SELADRS CNTLX from Cache. This signal allows the RH70 to gate its address and control information to Cache. Thus, the bus address, bus address extension, and the C0, C1 and CX control lines are gated onto the MBCBUS. C1 and C0 indicate a read from memory and CX is specified as 0.

After gating the address and control onto the MBCBUS, the RH70 waits for REQ ACKN from Cache. This signal indicates that Cache has acknowledged the request and this controller will perform the next memory cycle. REQ ACKN clears the CNTLX REQ signal, which informs Cache that the REQ ACKN has been received by the RH70. On the trailing edge of REQ ACKN, the bus address, bus address extension and word count are incremented since the address and control information has been stored in Cache by this time.

The RH70 then waits for ADDR ACK, which, in a write command, has no significance other than to let the RH70 know that memory has responded to Cache. If ADDR ACK is not received by the Cache within 10 $\mu$s, Cache sends a Timeout signal to the RH70. If the Timeout signal is asserted and the Address Parity Error interface line is also asserted, Address Parity Error (APE bit 15-CS3), (PE bit 13-CS2) and Transfer Error (TRE bit 14-CS1) are set. If the Timeout signal is asserted and there is no address parity error, Non-Existent Memory (bit 11-CS2) and Transfer Error are both asserted.

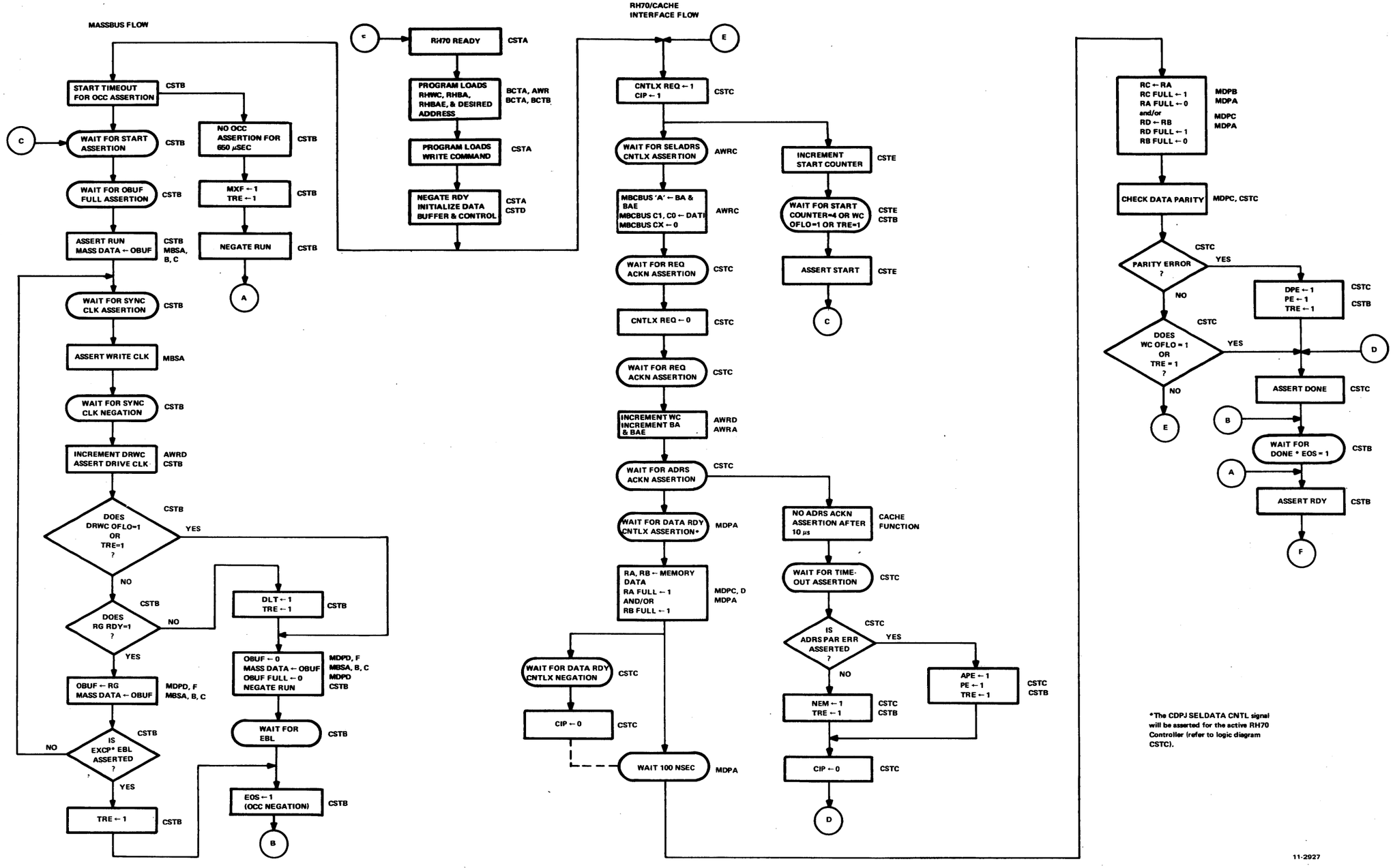



Figure 3-13 Write Command Flow Diagram

In both cases just described, the Cycle in Progress (CIP) flip-flop is cleared and the flow branches to connector D where DONE is asserted. DONE is asserted upon termination of the RH70/Cache Interface flow. Termination of the Massbus flow is indicated by End-of-Segment (EOS). When both flows are completed, the RH70 returns to the Ready state to accept another command.

Assume that there is no error condition and ADDR ACK has been received within 10 μs. The RH70 now waits for a DATA RDY CNTLX signal from Cache, indicating that memory has sent a data word. Upon receipt of DATA RDY CNTLX from Cache, the data is strobed into the RA and/or RB registers in the data buffer, and the RA FULL and/or RB FULL flip-flops are set. If a double-word operation is specified, both the RA and RB registers are loaded. If a single-word operation is specified, either RA or RB is loaded. A 100-ns delay is initiated to allow time for checking the parity on the data as it is strobed from RA into RC and/or from RB into RD.

If the word is an even word, the data is clocked from RA into RC, RA FULL is cleared, and RC FULL is set. If the word is odd, the data is clocked from RB into RD, RB FULL is cleared, and RD FULL is set. If a double-word operation is specified, the contents of RA and RB are both transferred. If a parity error is detected on the data as it is being strobed into RC and/or RD, Data Parity Error (DPE OW bit 14 or DPE EW bit 13 - CS3), Parity Error (PE bit 13–CS2), and Transfer Error (TRE bit 14–CS1) are set. This causes the assertion of DONE to terminate the RH70/Cache Interface flow. With no error conditions present, the flow loops back to the CNTLX REQ which initiates another memory cycle. This process continues until the number of words specified by the Word Count register have been transferred from memory to the Data Buffer, or until an error condition occurs. In either case, DONE is asserted and the flow branches to connector B, where it waits for the Massbus flow to complete. The completion of both flows returns the RH70 to the RDY state.

### 3.5.2 Massbus Flow

When the command is loaded by the program and the RH70 becomes busy, the Timeout for the assertion of OCC (Occupied) is initiated. The assertion of OCC indicates that the drive has recognized the command and is able to execute it. If the drive does not respond to OCC within 650 μs, a Missed Transfer error (MXF) is asserted and RUN is negated (if it was asserted). Also, the RH70 becomes Ready and waits for a new command.

The RH70 awaits the assertion of START, which occurs when the start counter has a count of four, word count overflow occurs, or Transfer Error is asserted. The RH70 then waits until a word in the data buffer has sequenced to OBUF. At this time, the RUN line is asserted on the Massbus and the contents of OBUF is placed on the Massbus data lines. The assertion of RUN signals the Massbus device to begin processing the data word. Upon completion, the device issues SCLK (Sync Clock) requesting another data word. The RH70 reroutes this to the drive as WCLK (Write Clock) with the next data word.

When SYNC CLK is negated, drive word count is incremented. Drive word count keeps track of the number of Massbus cycles. In addition to incrementing drive word count, the negation of SYNC CLK creates a 50-ns pulse, designated DRIVE CLK.

If drive word count overflow occurs, or an error condition is posted (TRE), the OBUF register is cleared and 0s are written onto the remainder of the record. Also, RUN is negated and the OBUF FULL flip-flop is cleared. If frame count in the drive is set to less than twice the word count, frame count overflow will occur (EXCP asserted) and may cause a Frame Count Error (FCE). If frame count is set to more than twice the word count, word count will overflow first, negating the RUN line. The RH70 now waits for End of Block (EBL) from the drive. Since RUN is negated at this time, End of Segment (EOS) is asserted, terminating the Massbus flow. The flow branches to connector B to wait for termination of the RH70/Cache Interface flow. When both flows are complete, the RH70 goes to the Ready state to be able to accept another command.

If there is no error or drive word count overflow, the above described path is bypassed and the flow sequences down to the block designated "Does RG RDY = 1?" If RG RDY is a 0 (unasserted) indicating that a word is not available to be loaded into OBUF, a Data Late (DLT bit 15 of CS2) error is posted, Transfer Error is posted, the OBUF register is cleared, and the remainder of the segment is zero-filled.

If RG RDY is asserted, indicating that a word is available in RG, the word in RG is transferred to OBUF. The contents of OBUF is directly gated onto the Massbus data lines. If both EXC (asserted due to a drive error) and EBL are asserted, Transfer Error is set. At the time of EBL and EXC there are no more SYNC CLK signals from the drive, and the RH70 thus asserts EOS. The flow branches to connector B to wait for termination of the RH70/Cache Interface flow.

If there was no drive error, the flow loops back to the point where the RH70 waits for the assertion of SYNC CLK, since the data has been changed and a new word is available for transfer. This process continues until an error is asserted or drive word count overflow occurs, both of which cause the operation to terminate.

## 3.6 READ COMMAND FLOW DIAGRAM DESCRIPTION

The flow diagram for the Read command is shown in Figure 3-14. This paragraph describes the read command, where data is read from the drive and transferred to memory via the RH70 data buffer. The RH70 must be in the Ready state (indicating that it is not doing a data transfer command). The assertion of bit 07 (RDY) in the CS1 register puts the RH70 in the Ready state. The program loads the word count, bus address, bus address extension registers and the frame count.

The read command is then loaded which negates the RDY bit (RH70 becomes busy) and causes the data buffer and control to be initialized.

At this point, the flow divides into two major flows: 1) a Massbus flow, which causes data words from the drive to be transferred to the RH70 data buffer, and 2) an RH70/Cache Interface flow, which transfers words from the RH70 data buffer to memory. Both flows operate asynchronously until both DONE and EOS (End of Segment) are asserted (see connector B of right of flow diagram). DONE indicates termination of the RH70/Cache Interface flow and EOS indicates termination of the Massbus flow. When both flows are terminated, RDY is asserted to allow a new command to be loaded. Paragraph 3.6.1. describes the Massbus flow; While Paragraph 3.6.2 describes the RH70/Cache Interface flow.

### 3.6.1 Massbus Flow

When the Read Command is loaded by the program and the RH70 becomes busy the timeout for the assertion of OCC (occupied) is initiated from the drive. The assertion of OCC indicates that the drive has recognized a data transfer command and is able to execute it. If OCC is not received from the drive within 650 $\mu$s a missed transfer error (MXF bit 9–CS2) and Transfer Error (TRE bit 14–CS1) are asserted, RUN is negated (if it was asserted) and the flow branches to connector A where the RH70 returns to the Ready state.

The RH70 asserts the RUN line, indicating to the drive that it is ready to accept data. When the Massbus device is ready to begin transferring data, it places the data on the Massbus and sends it to the RH70, accompanied by SCLK. On the negation of SCLK, the RH70 increments the drive word count which keeps track of the number of Massbus cycles. The Massbus device waits for the next two characters from the Drive, assembles them into a 16-bit word and sends it into the RH70, accompanied by SCLK. This process continues until an error condition is posted, word count overflow is detected, or the interrecord gap is detected in the drive. On the trailing edge of SCLK, the RE register is examined. If the register is full (RE FULL=1), Data Late (DLT bit 15-CS1) and Transfer Error (TRE bit 14-CS1) conditions are asserted, since there is no place to store the data word from the Massbus. The RUN line is negated and at EBL time, EOS (End of Segment) is asserted terminating the Massbus flow. The flow branches to connector B to wait for termination of the RH70/Cache Interface flow. When both flows complete, the RH70 returns to the Ready state.

If the RE register was empty, the data from the Massbus is strobed into it and the RE FULL flag is asserted. Should an error condition be asserted, or drive word count overflow occur at this time, the RUN line is negated and the RH70 waits for EBL, ignoring any data words prior to EBL. With RUN negated at EBL time, EOS is asserted to terminate the Massbus flow. The flow branches to connector B, where it waits for the RH70/Cache Interface flow to complete, so the RH70 may return to the Ready state.

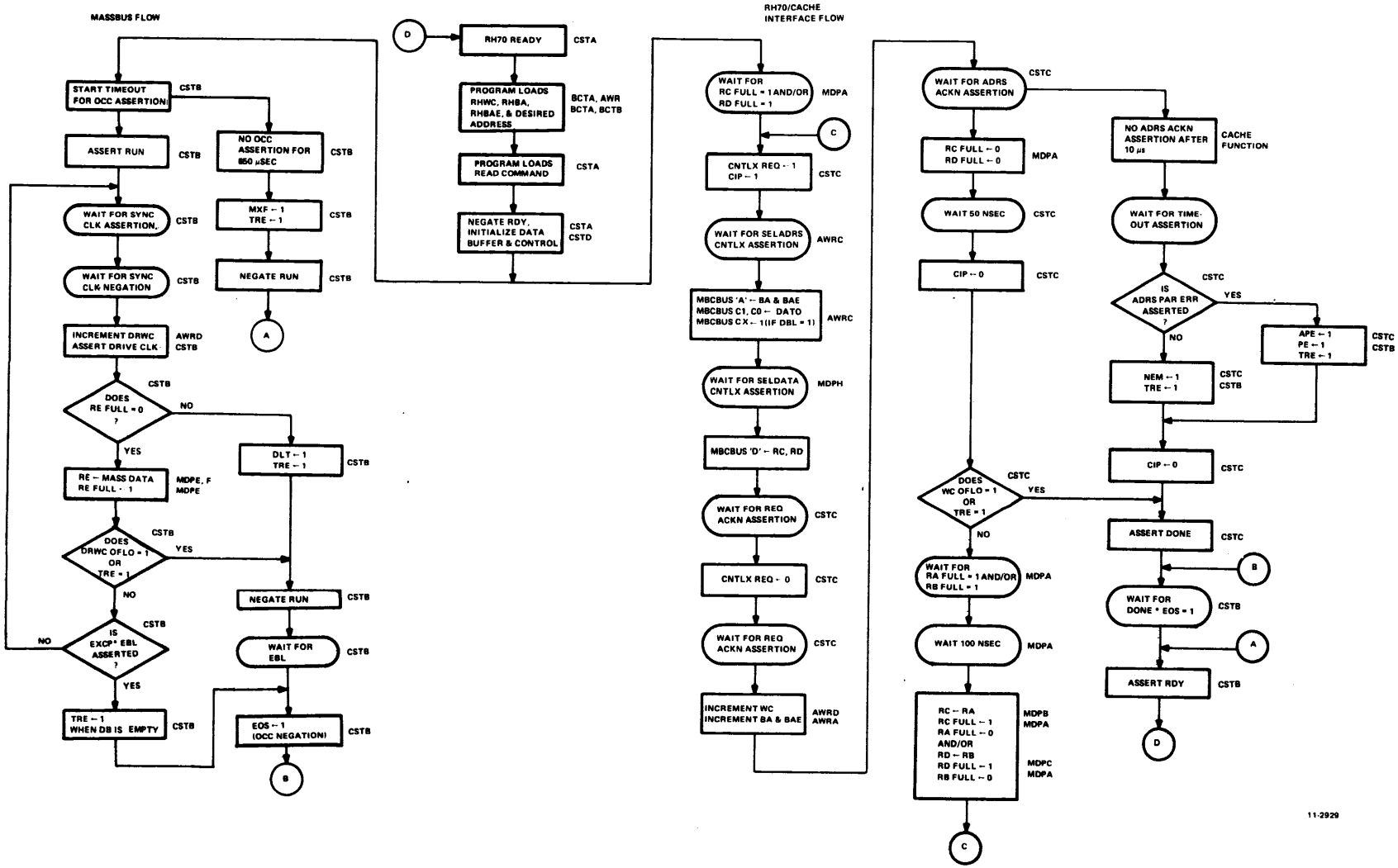



Figure 3-14 Read Command Flow Diagram

However, if no error occurs, or if drive word count overflow does not occur, the EXC line is monitored. If EXC is asserted (denoting a drive error) at EBL time, the transfer error bit is set after the RH70 data buffer has been emptied. This allows the RH70/Cache Interface flow to transfer all data words currently in the data buffer to memory, prior to asserting the error condition. The RH70 terminates the Massbus flow and the flow branches to connector B to wait for completion of the RH70/Cache Interface flow.

If there was no drive error, the flow loops back to the point where the RH70 waits for the assertion of SYNC CLK and the next data word is loaded into the data buffer. This process continues until an error is asserted, or until drive word count overflow occurs, both of which cause the data transfer to terminate.

### 3.6.2 RH70/Cache Interface Flow

The RH70/Cache Interface flow is not initiated until the data words from the drive have sequenced through the data buffer and into RC and/or RD via the Massbus flow. When a data word reaches RC and/or RD, the RH70 sends a CNTLX REQ signal to Cache, requesting a memory cycle. In addition, the RH70 sets the CIP flip-flop, indicating that a cycle is in progress.

The RH70 now waits for Cache to acknowledge the memory request by issuing SELADRS CNTLX. This signal allows the RH70 to gate its address and control information onto the address and control lines of the MBCBUS, respectively. The bus address and bus address extension is gated onto the address lines and a DATO operation is gated onto the C0 and C1 control lines. If a double-word transfer has been specified, the CX line is asserted; if a single-word operation is specified, the CX line is unasserted.

At this point, the RH70 waits for a SELDATA CNTLX signal from Cache. Upon receipt of this signal, the RH70 gates the data from RC and/or RD onto the MBCBUS data lines. After the address, control, and data have been transferred to Cache, the RH70 waits for REQ ACKN, indicating that Cache has acknowledged the memory request, accepted the address and control, and is about to start a Main Memory cycle. The assertion of REQ ACKN causes the CNTLX REQ to drop. On the negation of REQ ACKN, word count, bus address and bus address extension are incremented since by this time, Cache has accepted and stored the address and control information.

The RH70 now waits for the ADRS ACK signal from Cache. Cache has a timeout feature which causes a Timeout signal to be asserted to the RH70 if ADRS ACK is not received within 10 $\mu$s. If the Timeout is asserted (indicating no ADRS ACK), the Address Parity Error line is monitored. If there is an address parity error, Address Parity Error (APE bit 15–CS3), Parity Error (PE bit 13–CS2), and Transfer Error (TRE bit 14–CS1) are all asserted. If there is no address parity error, the timeout is due to non-existent memory and NEM (bit 11 in CS2) is asserted along with Transfer Error. In both cases, the CIP flip-flop is cleared and DONE is asserted, terminating the RH70/Cache Interface flow. When EOS occurs to terminate the Massbus flow, the RH70 returns to the Ready state.

Assume that ADRS ACK was received within the 10 $\mu$s Timeout. The receipt of ADRS ACK, during a Read command, indicates that Main Memory has received the data and now the RC FULL and RD FULL flip-flops can be cleared. After a 50-ns delay, the CIP flip-flop is cleared, indicating termination of the memory cycle. If there is an error condition, or if word count overflow occurs, DONE is asserted to terminate the RH70/Cache Interface flow. However, if there is no error and if word count has not overflowed, the RH70 waits for the RA or RB register to be loaded and the corresponding RA FULL or RB FULL flag to be asserted. If a double-word operation is specified, both RA and RB will be loaded and the RA FULL and RB FULL flags are both asserted.

The RH70 then waits 100 ns and transfers the contents of RA into RC and/or RB into RD and sets and clears the appropriate flags. For example, if a double-word transfer is specified, the contents of RA is transferred into RC, the RA FULL flag is cleared and the RC FULL flag is set. Also, the contents of RB is transferred into RD, the RB FULL flag is cleared and the RD FULL flag is set. The flow now loops back to the point where the RH70 makes another request to memory (CNTLX REQ). This request will transfer the contents of RC and/or RD into memory. This process continues until there is an error, or until word count overflow occurs. Either condition causes the operation to terminate, as previously described.

# CHAPTER 4
# DETAILED LOGIC DESCRIPTION

## 4.1 GENERAL

This chapter provides a detailed description of the RH70 Massbus Controller logic diagrams. These diagrams should be used in conjunction with the command flowcharts in the associated Massbus device manual to provide both an overall and detailed understanding of the RH70. The diagrams described in this chapter are tabulated in Table 4-1.

The M8150 module is designated MDP and contains the data buffer and the associated parity checking and generating logic. The M8151 module is designated CST and contains the control and status logic. The M8152 module is designated AWR and contains the address and word count register logic. The M8153 module is designated BCT and contains the bus control logic.

Detailed timing diagrams are also included in the chapter and may be used in conjunction with the detailed logic descriptions to show timing relationships between signals. The timing diagrams included are listed below:

Figure 4-1 Read or Write-Check Massbus Timing Diagram

Figure 4-2 Write Massbus Timing Diagram

Figure 4-3 Data Buffer Maintenance Operation Timing Diagram

Figure 4-4 Data Buffer Write Command Timing Diagram

Figure 4-5 Data Buffer Read Command Timing Diagram

Figure 4-6 Data Buffer Write-Check Command Timing Diagram

**Table 4-1**
**Listing of RH70 Logic Diagrams**

| Logic Print | Functions |
|---|---|
| BCTA | Register Selection |
| BCTB | SSYN and Demand Logic |
| BCTC | Unibus Interrupt Logic |
| BCTD | Unibus Data Transceivers |
| AWRA | Bus Address Reg (11:01) |
| AWRB | Bus Address and Extension (21:12) and Control Parity Out |
| AWRC | BUSI MUX (BA and BAE Registers) and Address Drivers |
| AWRD | Word Count Register (07:00) |
| AWRE | Word Count Register (15:08) and WC OFLO |
| AWRF | BUSI MUX (WC and Remote Registers) and MCPE |
| MDPA | Memory Register Control |
| MDPB | Even Word Memory Data Registers and Mixer |
| MDPC | Odd Word Memory Data Registers and Memory Parity Generator/Checkers |
| MDPD | Data Registers Control |
| MDPE | Write-Check Logic, Clock RE and Data Parity |
| MDPF | Data Buffer Registers |
| MDPH | IMX and Data Drivers |
| MDPJ | Data Path Block Diagram |
| CSTA | CS1, CS2, and CS3 Control and Status, Function Select |
| CSTB | Errors, IE, Intr. Req. and Massbus Control |
| CSTC | PE, NEM, and Memory Cycle Control |
| CSTD | BUSI MUX (CS1 and CS2 Registers) and Clear Logic |
| CSTE | BUSI MUX (DB and CS3 Register) and START Logic |
| MBSA | Massbus Transceiver (Massbus Cable A) |
| MBSB | Massbus Transceiver (Massbus Cable B) |
| MBSC | Massbus Transceiver (Massbus Cable C) |

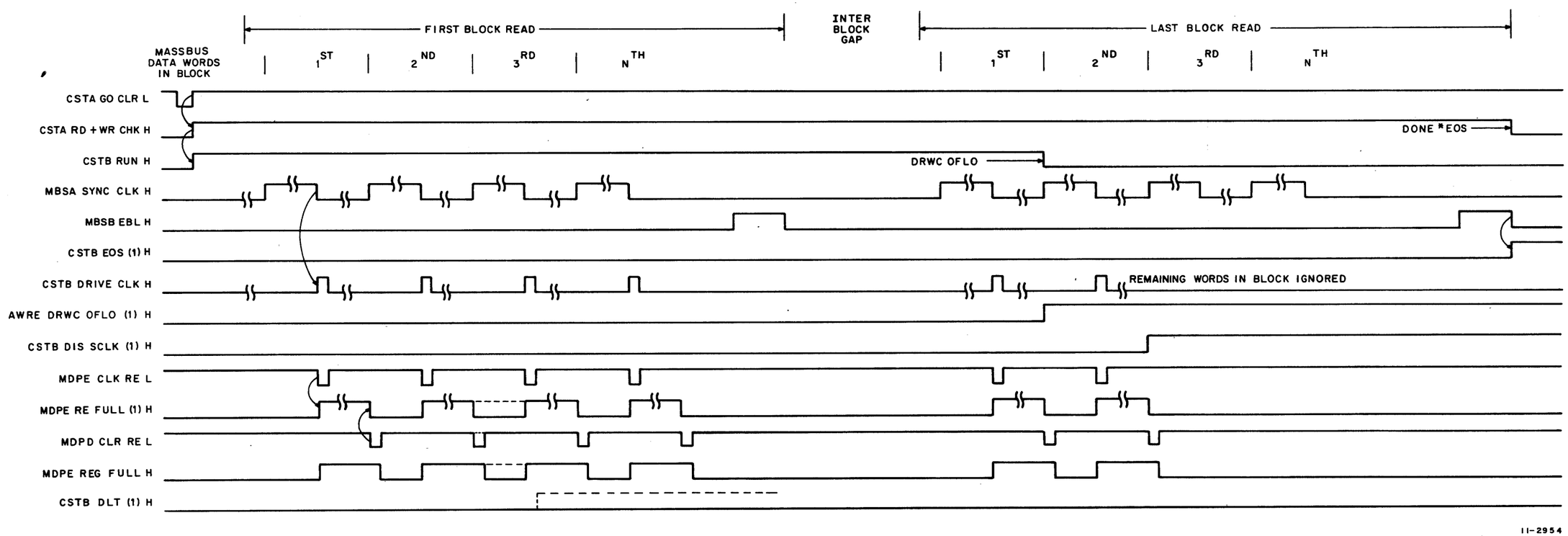



Figure 4-1 Read or Write-Check Massbus Timing Diagram

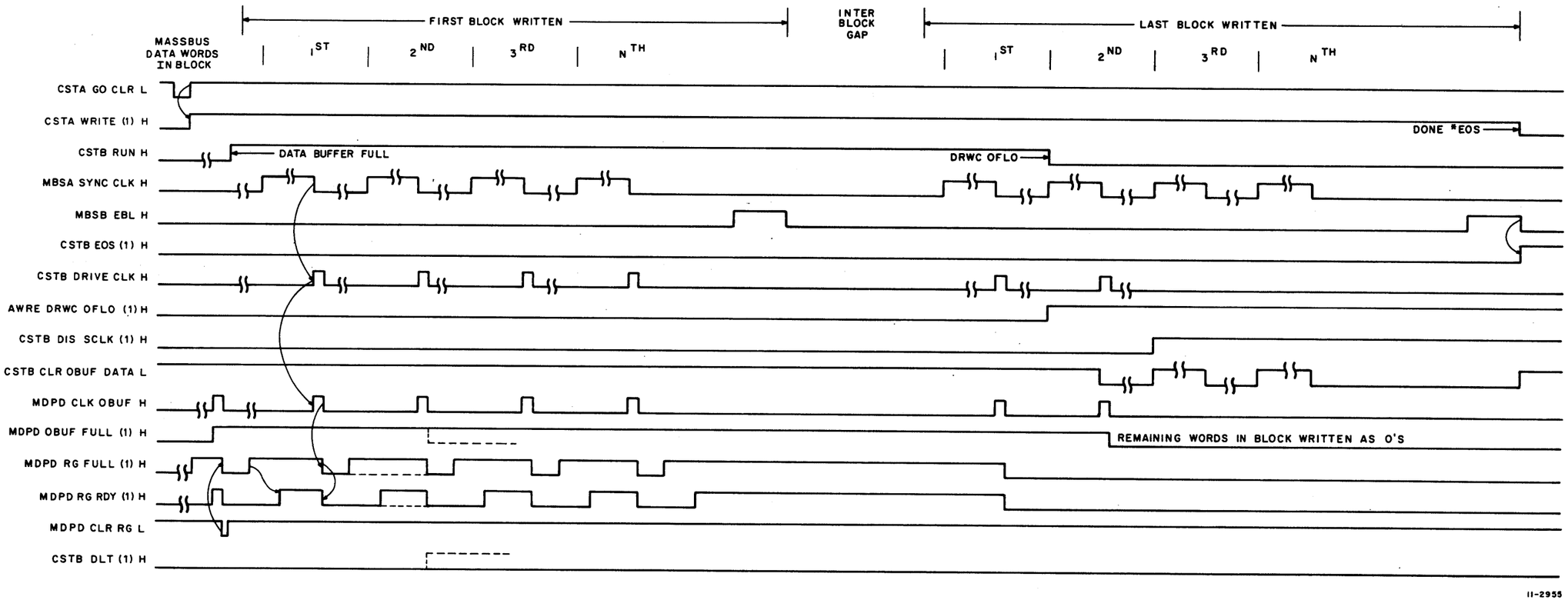



Figure 4-2 Write Command Massbus Timing Diagram

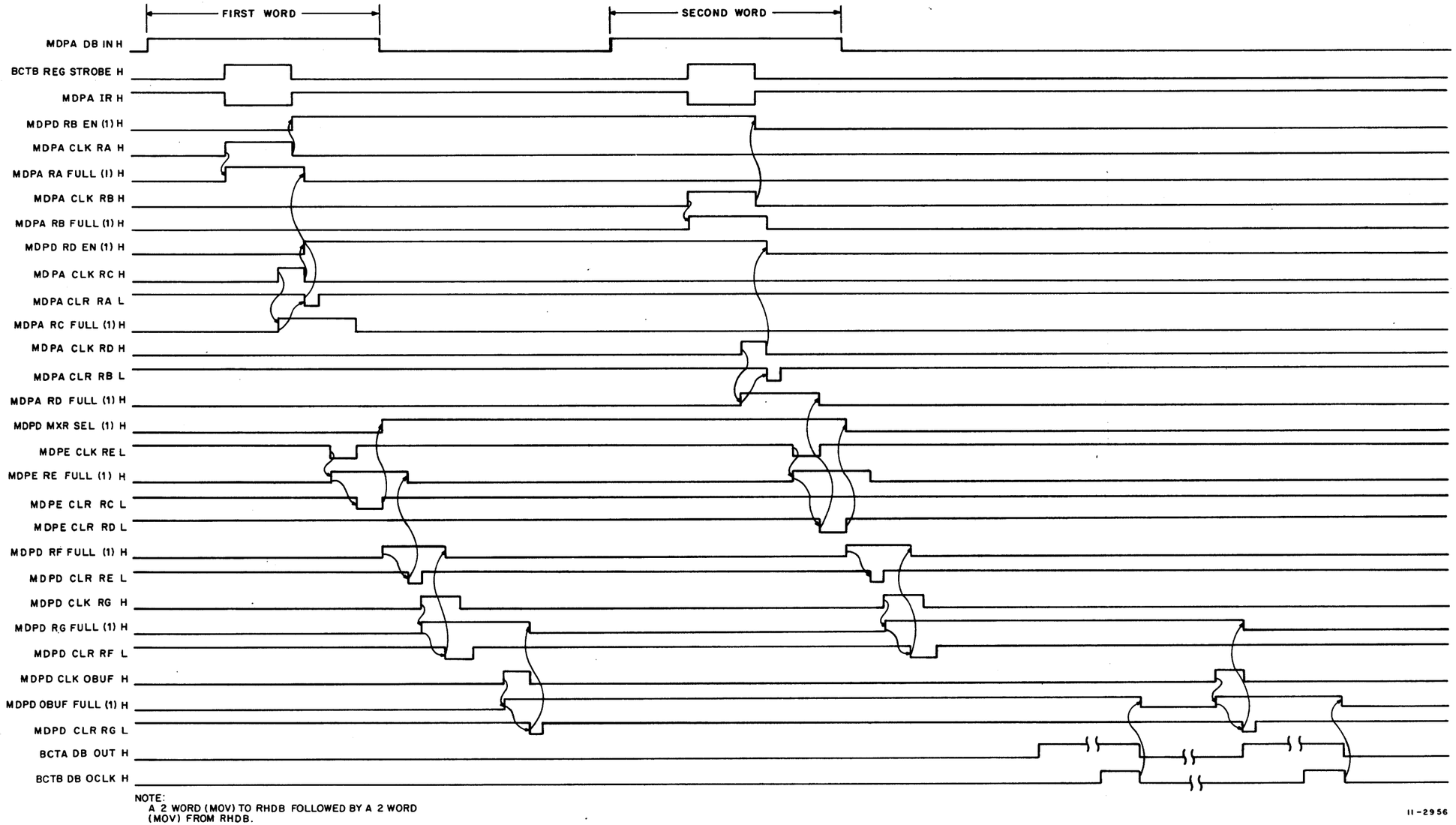



Figure 4-3 Data Buffer Maintenance Operation Timing Diagram

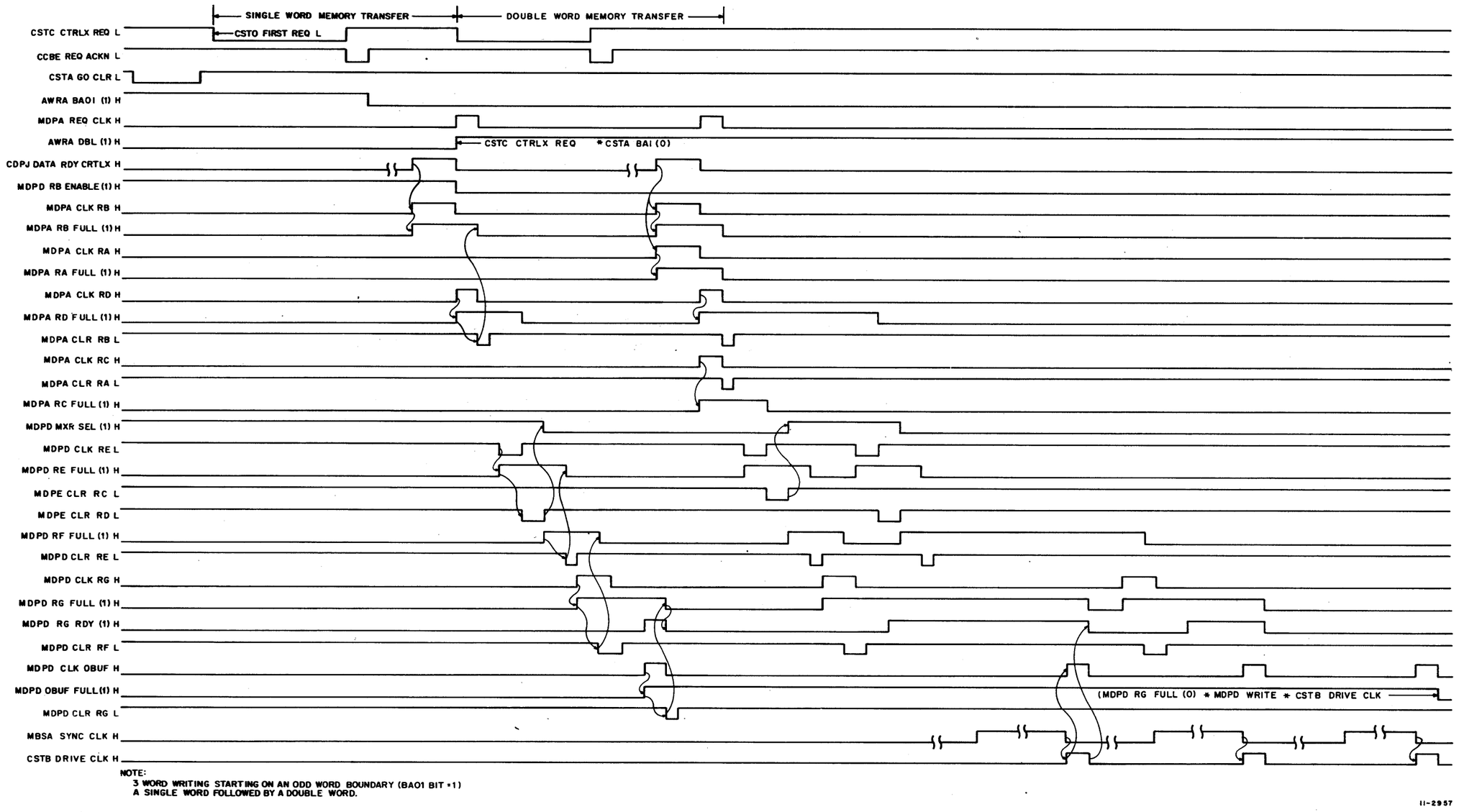



Figure 4-4 Data Buffer Write Command Timing Diagram

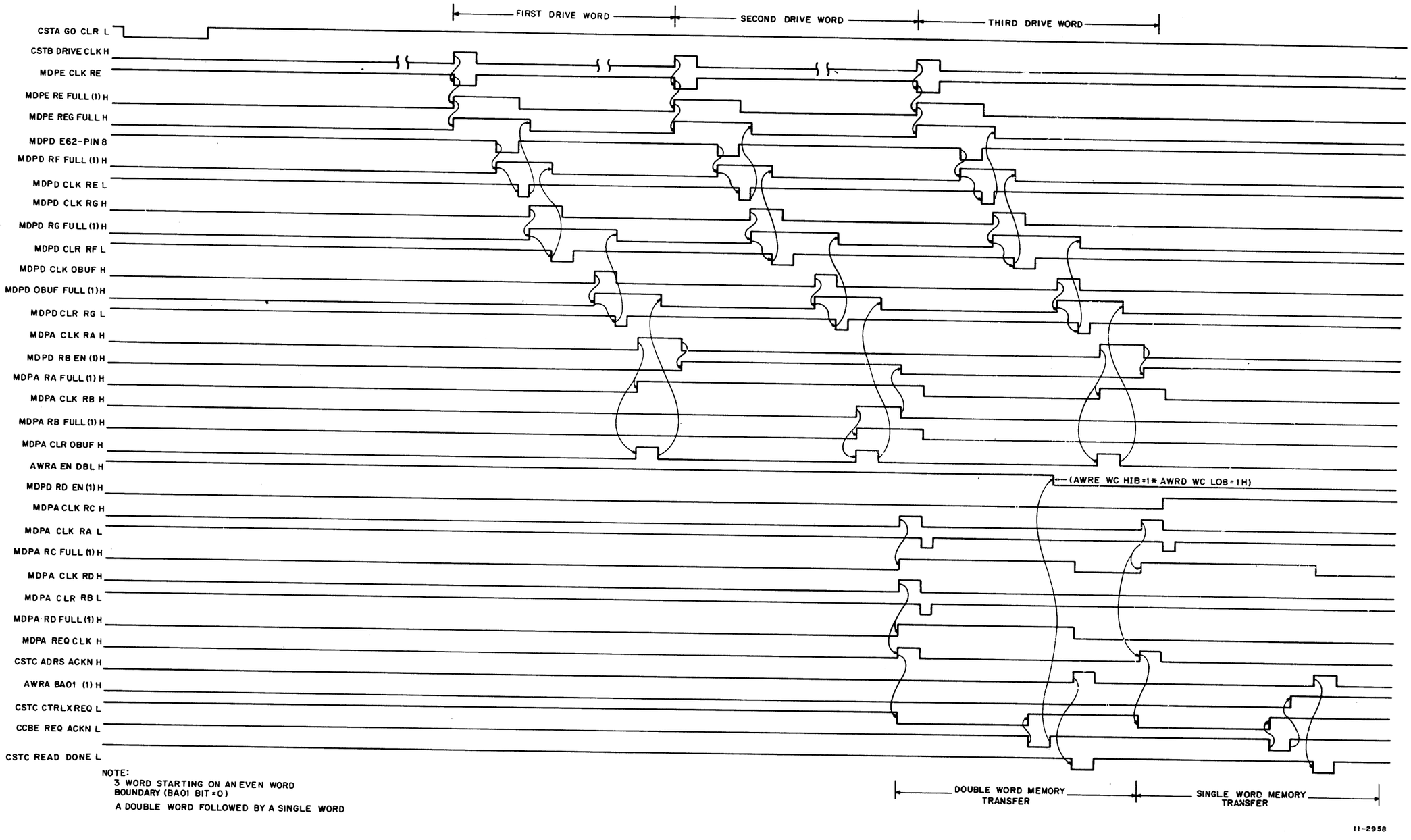



Figure 4-5 Data Buffer Read Command Timing Diagram

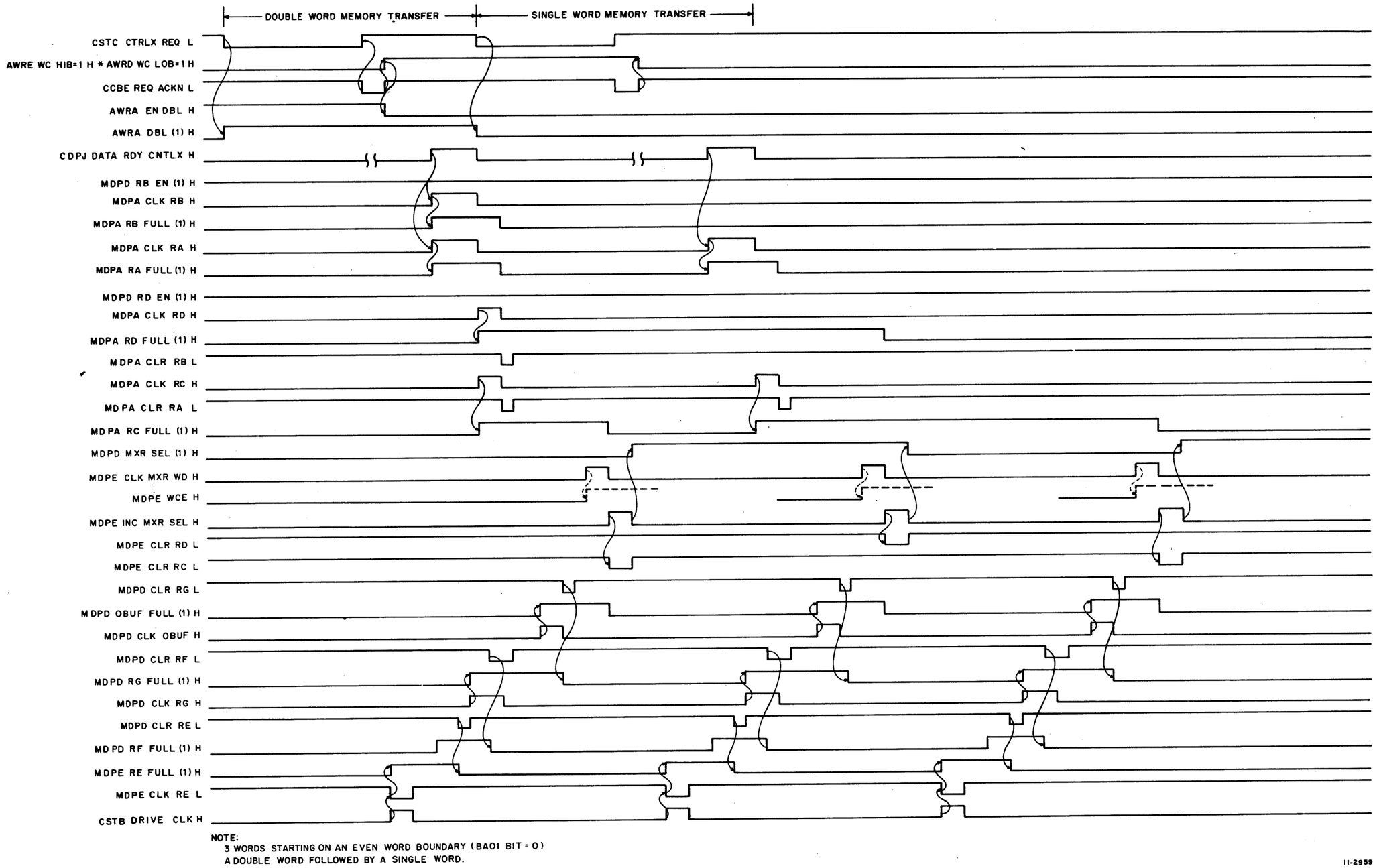



Figure 4-6 Data Buffer Write-Check Command Timing Diagram

## 4.2 BCTA LOGIC DIAGRAM

This diagram contains the register selection logic used by the program to select local RH70 registers or remote registers in the associated drive. The register address is supplied to 15 Unibus receivers (8640) via the Unibus. Bits 17 - 13 of the register address are asserted designating the I/O area. Bits 12 - 6 are fed to a series of jumper Exclusive-OR gates, whose outputs are collector-ORed to compare the device address against the Unibus address that is received. This detects if the program is addressing this device. If any of the outputs of these gates go low, it forces the output line low, as is the case where the Unibus address does not match the selected address of the RH70. The addresses to which the RH70 responds can be relocated by modifying the jumpers. If a jumper is left in, it represents a logic 0; if it is cut, it represents a logic 1. The register address bits are asserted low on the Unibus. For example, address bit 12 is low at the input to the 8640 Unibus receiver. The output of this gate goes high. This is compared to the jumper in place which is low. The output of the Exclusive-OR gate, after inversion, is low and this drives the collector-ORed output line low to inhibit DEV SEL. On the other hand, if the jumper is out (representing a 1), the Exclusive-OR gate compares two high inputs, yielding a high output which enables the DEV SEL signal for that bit.

The BAE and CS3 registers "float" over the Massbus addresses; their addresses will vary based on the number of subsystem registers. These registers are logically defined as the last two registers in the system. The following paragraphs describe how these registers are addressed.

Bits 4 - 1 of the Unibus address are supplied to a 32-cell Read-Only Memory (ROM). These bits specify one of 16 registers in a subsystem. In subsystems with 16 registers or less, address bit 05 (BUS A05 L) is applied to the exclusive-OR gates to decode the device address; otherwise, there would be two fields where a register address might be located (one field where bit 05 is a 1 and one field where bit 05 is a 0). In subsystems with more than 16 registers, bit 05 is applied to a ROM (EO50) where it is used with bits 04 - 01 to specify one of 32 registers (Figure 4-7). For example, if all the inputs to the ROM are 1s (corresponding to an octal address of 37), the ROM maps this to a Massbus address of 33. This corresponds to a Unibus address of the base address (defined as the address of the first register) plus 76, since Unibus addresses are incremented by 2. The ROM also contains a BCTA LOCAL REG H signal, which specifies a local (RH70) register when asserted and specifies a remote register when unasserted. The M6 and M7 outputs of the ROM are used to select additional registers within the RH70.

Address bits 04 - 02, in addition to feeding the ROM, are input to Comparator E046 where they are compared with the register length field. The register length field is equal to the weighted value of the jumpers removed, plus 2. If more than 16 registers are specified, address bit 05 is also applied to the comparator. If the subsystem contains a total of 16 registers for example, (as in the TWU45 subsystem) the jumpers corresponding to a value of 8 (pins 7 and 10), 4 (pins 6 and 11) and 2 (pins 5 and 12) are cut. This yields 14 plus 2 or 16. The "plus 2" is to allow for the CS3 and BAE registers which represent the last two registers in a subsystem.

The comparator has three outputs: A=B, A>B and A<B; only one of these can be asserted at any given time. Each is described below.

A=B - The A=B output is asserted if the CS3 or BAE register is specified, which is defined as the end of the field of registers for this subsystem (register length field = incoming address). This output, in conjunction with the state of address bit 01 (BUS A01 L), is applied to BCD-to-Decimal Decoder E010 in order to determine whether the CS3 or BAE is addressed. If address bit 01 is asserted, the CS3 register is specified and if this bit is negated, the BAE register is specified. Note that BCTA C1 L and BCTA DEV SEL L are also applied to the decoder. With C1 L asserted, it indicates that a write operation is to be performed; with C1 L unasserted, it indicates that a read operation is to be performed to the register. The DEV SEL signal ensures that this is the device that has been selected. As an example of the operation, assume that a subsystem has been cut for 14 registers (the 8 and 4 jumpers being removed from E041) and that the program specifies a write into the BAE register. The A=B output of the comparator is asserted and address bit 01 will be negated. These signals are fed to decoder E010, along with C1 L asserted, causing the decoder to assert BCTA BAE IN L.

A>B - The addresses of the CS3 and BAE registers are variable, depending on the number of registers employed in the subsystem. They will be assigned some address between the base address +22 and

REGISTER SELECT PROM – 23040A1 (8223)

| OCTAL ADDRESS | M7 | M6 | LOCAL REG M5 | RSEL4 M4 | RSEL3 M3 | RSEL2 M2 | RSEL1 M1 | RSEL0 M0 |
|---|---|---|---|---|---|---|---|---|
| 00 | L | L | L | L | L | L | L | L |
| 01 | H | L | H | L | L | L | H | H |
| 02 | H | H | H | L | L | L | L | L |
| 03 | H | H | L | L | L | H | L | H |
| 04 | H | H | H | L | L | L | H | L |
| 05 | H | H | L | L | L | L | L | H |
| 06 | H | H | L | L | L | L | H | L |
| 07 | L | H | L | L | L | H | L | L |
| 10 | H | H | L | L | L | H | H | H |
| 11 | H | H | H | L | L | L | L | H |
| 12 | H | H | L | L | L | L | H | H |
| 13 | H | H | L | L | L | H | H | L |
| 14 | H | H | L | L | H | L | L | L |
| 15 | H | H | L | L | H | L | L | H |
| 16 | H | H | L | L | H | L | H | L |
| 17 | H | H | L | L | H | L | H | H |
| 20 | H | H | L | L | H | H | L | L |
| 21 | H | H | L | L | H | H | L | H |
| 22 | H | H | L | L | H | H | H | L |
| 23 | H | H | L | L | H | H | H | H |
| 24 | H | H | L | H | L | L | L | L |
| 25 | H | H | L | H | L | L | L | H |
| 26 | H | H | L | H | L | L | H | L |
| 27 | H | H | L | H | L | L | H | H |
| 30 | H | H | L | H | L | H | L | L |
| 31 | H | H | L | H | L | H | L | H |
| 32 | H | H | L | H | L | H | H | L |
| 33 | H | H | L | H | L | H | H | H |
| 34 | H | H | L | H | H | L | L | L |
| 35 | H | H | L | H | H | L | L | H |
| 36 | H | H | L | H | H | L | H | L |
| 37 | H | H | L | H | H | L | H | H |

| REGISTER SELECTED** | UNIBUS ADDRESS |
|---|---|
| RHCS1 MASSBUS (00) | BASE ADDRESS |
| RHWC | BASE +2 |
| RHBR | BASE +4 |
| MASSBUS (05) | BASE +6 |
| RHCS2 | BASE +10 |
| MASSBUS (01) | BASE +12 |
| MASSBUS (02) | BASE +14 |
| MASSBUS (04) | BASE +16 |
| MASSBUS (07) | BASE +20 |
| RHDB | BASE +22 |
| MASSBUS (03) | BASE +24 |
| MASSBUS (06) | BASE +26 |
| MASSBUS (10) | BASE +30 |
| MASSBUS (11) | BASE +32 |
| MASSBUS (12) | BASE +34 |
| MASSBUS (13) | BASE +36 |
| MASSBUS (14) | BASE +40 |
| MASSBUS (15) | BASE +42 |
| MASSBUS (16) | BASE +44 |
| MASSBUS (17) | BASE +46 |
| MASSBUS (20) | BASE +50 |
| MASSBUS (21) | BASE +52 |
| MASSBUS (22) | BASE +54 |
| MASSBUS (23) | BASE +56 |
| MASSBUS (24) | BASE +60 |
| MASSBUS (25) | BASE +62 |
| MASSBUS (26) | BASE +64 |
| MASSBUS (27) | BASE +66 |
| MASSBUS (30) | BASE +70 |
| MASSBUS (31) | BASE +72 |
| MASSBUS (32) | BASE +74 |
| MASSBUS (33) | BASE +76 |

*Address bit 05 may be grounded if there are less than 16 registers in the subsystem.

**The BAE and CS3 registers are assigned Massbus addresses which may vary depending on the number of registers employed in a particular subsystem. These registers are respectively located at addresses 772070 and 772072 for the RWS04 subsystem; 776750 and 776752 for the RWP04 subsystem, and 772474 and 772476 for the TWU16 subsystem.

Figure 4-7 Mapping Massbus/Unibus Addresses

the base address +76 (Figure 4-7). Note that this area is in the Massbus register field and since these registers are local, it is necessary to inhibit a Massbus control bus cycle when either of these registers is specified.

The A>B output of Comparator E046 is asserted when the register length field is greater than the Unibus address specified. This means that a valid register has been addressed and causes the Comparator to assert BCTA REM ENA H. This signal is ANDed with BCTA LOCAL REG H in NOR gate E030-pin 1. If a local register has been designated, the gate is inhibited and BCTA REMOTE REG H is negated, inhibiting the Massbus cycle. If a remote register is designated, the gate is enabled and BCTA REMOTE REG H is asserted to allow the Massbus cycle to be initiated.

A<B - The BCTA ILLEGAL REG H output of the Comparator is asserted when A<B (that is, when the Unibus address specified is greater than the register length field). This condition indicates that an illegal register has been addressed and this device is prevented from being selected by negating the BCTA DEV SEL signal in AND gate E038-pin 12.

### 4.2.1 Local/Remote Register Selection

In addition to encoding the register address, the ROM asserts the BCTA LOCAL REG H signal if a local register is being accessed. The Massbus handshaking sequence, necessary to access a register located in a drive, is inhibited in this case. If a remote register is being addressed, the BCTA LOCAL REG H signal is not asserted (the Massbus handshaking sequence is enabled) and the DEMAND signal is initiated.

NOTE

**The CS1 register is shared by the RH70 Controller and the associated drive. The LOCAL REG H signal is not asserted to address this register. If LOCAL REG H is asserted, the Massbus handshaking sequence is inhibited. This would prevent access to the portion of the CS1 register located in the drive.**

### 4.2.2 RSEL Signals

The BCTA RSEL 04 - RSEL 00 signals from the output of the ROM are supplied to the Massbus, which will be decoded in the drive to select a specific register. In addition, RSEL 00 and RSEL 01 are supplied to a 7442-BCD-to-decimal decoder (E019) and are used when BCTA LOCAL REG H signal is asserted. The decoder decodes the CS2, DB, and BA registers and whether an input (from processor to RH70) or output (from RH70 to processor) function is to occur.

### 4.2.3 Decoder Inputs

Inputs D0, D1, and D2 to decoder E019 specify one of 8 outputs from f0 - f7 (outputs f8 and f9 are not used). RSEL 00 and RSEL 01 are applied to inputs D0 and D1 of the decoder, respectively, and specify one of the three above mentioned registers. BCTA C1 L is applied to input D2 of the decoder and specifies an input or output function. When BCTA C1 L is asserted, the D2 input to the decoder is low, enabling the register codes on outputs f0, f1, f2, and f3. This indicates a DATO or DATOB where the data is transferred from the master (processor) to the slave (RH70). The register signal names which incorporate the word IN denote a "write register" operation. When BCTA C1 L is not asserted, the D2 input to the decoder is high, enabling the register codes on outputs f4, f5, f6, and f7. This indicates a DATI or DATIP where the data is transferred from the slave (RH70) to the master (processor). The signal names designated the word OUT denote a "read register" operation. Input D3 is asserted low if the BCTA LOCAL REG H signal indicates that a local register is addressed. DEV SEL H is asserted for both read register and write register operations to select this device.

### 4.2.4 Decoder Outputs

With input D3 low, outputs f0 - f7 are enabled. If input D3 is not asserted, the outputs from f0 - f7 are inhibited and the decoder outputs are switched to f8 and f9, which are not used. Since the RSEL 04 - RSEL 00 signals are not used for selecting a drive register when the BCTA LOCAL REG H signal is asserted, it is possible to redefine the two bottom bits of the ROM (RSEL 00 and RSEL 01), depending on whether the register specified is local or remote. If a remote register is specified, RSEL 00 and RSEL 01 are used with RSEL 02, RSEL 03 and RSEL 04 to decode the register. If a local register is specified, RSEL 00 and RSEL 01 are fed to decoder E019 to decode the CS2, DB, and BA registers.

Decoder E015 is similar to E019 with a few exceptions noted below.

Input D2 operates exactly the same as described for decoder E019.

ROM outputs M6 and M7 are applied to D0 and D1 of decoder E015 and are used to decode the WC, CS1, and AS registers.

Input D3 is enabled when DEV SEL L is asserted. BCTA LOCAL REG H is not required since the M6 and M7 outputs of the ROM are not dual-defined and may be used anytime.

### 4.2.5 Word or Byte Addressing

The logic network with the A0, C0, and C1 inputs determines whether word or byte addressing is required and whether an input or output function is occurring. The network implements the chart shown below.

| BUS C1 L | BUS C0 L | |
|---|---|---|
| H | H | DATI (read from device) |
| H | L | DATIP |
| L | H | DATO (write into device) |
| L | L | DATOB* |
| | | *If BUS A 00 L =H, low byte is specified. |
| | | *If BUS A 00 L = L, high byte is specified. |

### 4.2.6 Control Lines

The C1 line from the Unibus generates the BCTA CTOD H (direction of transfer) signal. When this signal is asserted, the direction of data transfer is from the RH70 to the drive register via the Massbus. When the signal is not asserted, the direction of data transfer is from the drive register to the RH70, and then to the Unibus to be made available to the program.

### 4.2.7 ODD BYTE L Signal

The BCTA ODD BYTE L signal is asserted when performing a DATOB to the high byte and is used to generate BCTB INH DEM H (Inhibit Demand). INH DEM prevents the low (even) byte of the CS1 register (located in the drive) from changing when the program is doing a byte operation to the upper (odd) byte. (This is necessary since the Massbus does not implement byte operations and writeable bits in the low byte of CS1 are located in the drive.)

### 4.2.8 Device Select (DEV SEL) Logic

The upper portion of BCTA shows the DEV SEL LOGIC. Bits 17 - 5 of the Unibus address are used to generate BCTA DEV SEL L when BCTA MSYN H occurs. MSYN is delayed from the address 150 ns to allow the address to be properly decoded by the jumper Exclusive-OR gates. The MSYN signal then keys the DEV SEL signal which starts the register strobing sequence.

### 4.2.9 Control Out (CNTL OUT) Signal

The BCTA CNTL OUT L signal is used when reading a remote register in the drive. CNTL OUT L is generated when the BCTA LOCAL REG H signal is not asserted (remote mode), DEV SEL is asserted and the BUSA C1 L signal is not asserted. This signal switches the multiplexer on logic diagram AWRF to gate the Massbus asynchronous data to the Unibus data lines, making the register information available to the program.

## 4.3 LOGIC DIAGRAM BCTB

This logic diagram contains the register strobe logic for loading local registers, and the SSYN and DEMAND logic. The logic for addressing a non-existent device is also contained on this diagram.

### 4.3.1 Loading Local Registers

The leading edge of the BCTA DEV SEL signal triggers one-shot multivibrator E006. The external components associated with this multivibrator are chosen to provide an 85-ns delay. Consequently, the one-shot produces a negative-going, 85-ns pulse from the 0 output. At the end of 85 ns, the positive-going transition triggers a second one-shot (E006) multivibrator. When this one-shot is triggered, it causes BCTB REG STR H to be asserted, via gate E002-pin 8. This signal is used for clocking local registers contained in the RH70.

### 4.3.2 Deskew of DEMAND Signal

The second one-shot multivibrator (E006) just described provides a delay of 150 ns. The 0 output of this one-shot produces a negative-going pulse, 150-ns in duration. The positive-going transition of this pulse (which occurs approximately 235 ns after MSYN) clocks the SET DEM (Set Demand) flip-flop set if the BCTB INH DEM H signal remains unasserted. When asserted, INH DEM H inhibits a Massbus cycle if:

1. The register address is a local register address (BCTA LOCAL REG H).

2. The BCTA REM ENA L signal is not asserted, which means that the CS3 or BAE register is being selected (local RH70 registers).

3. If the access is to the odd byte (BCTA ODD BYTE L) in the CS1 register (BCTA CS1 IN L). The CS1 register is shared by the RH70 and the drive, with the odd byte being in the RH70 and the even byte being in the drive. ODD BYTE L and CS1 IN L generate BCTB INH DEM H to inhibit the Massbus handshake sequence. This prevents altering the even (low) byte of the CS1 register located in the drive, when the program is doing a byte operation, to the odd (high) byte in the RH70. This is necessary because the Massbus cannot differentiate byte from word operations.

4. The STOP DEM L signal is asserted when the processor tries to load a function code, specifying a data transfer operation, into the drive while the RH70 is already busy executing a data transfer function with that drive or some other drive. For example, if unit 0 is doing a read data transfer and the processor tries to do a read or write data transfer in unit 1, the CSTB STOP DEM L signal prevents the function code from being transferred to unit 1; otherwise, there would be the OR condition of data from unit 0 and unit 1 on the synchronous Massbus and the program could not distinguish unit 0 data from unit 1 data.

With the SET DEM flip-flop set, the DEMAND flip-flop is set, provided the TRA from the previous Massbus cycle has completed. If the previous Massbus cycle has not completed, the BCTB TRA L signal prevents the DEMAND flip-flop from setting. The DEMAND signal is sent to the drive via the Massbus and requests a Massbus control bus cycle.

### 4.3.3 Register Strobe (REG STR)

A BCTB REG STR H signal is generated 85 ns after MSYN and is used for clocking the local registers.

### 4.3.4 Writing a Remote Register

The BCTB GATE CNTL H signal is used when writing a remote register in the drive. BCTB GATE CNTL H is generated when BCTA CTOD H (write) and BCTA DEV SEL H are asserted and gate the Unibus data signals to the Massbus control lines.

### 4.3.5 SSYN Logic

The SSYN logic determines when SSYN is sent to the Unibus. Each of the various methods of setting SSYN is described below.

1. Setting SSYN-Writing Remote Register (register electrically located in the drive) – When the program writes a remote register via the RH70, the unit select bits, the RSEL 04 – 00 signals, and the data are gated onto the Massbus. The unit select bits select the specified unit and the RSEL 04 – 00 signals select the appropriate register in that unit. If the RH70 is *writing* into a remote register, BCTA CTOD H is asserted. After a 235-ns deskew period, the RH70 asserts DEMAND on the Massbus. When the drive sees DEMAND and recognizes its own unit select code, it takes the data and issues MBSC TRA. The MBSC TRA signal is applied to one input of gate E001-pin 13, provided the Attention Summary register was not addressed. If the Attention Summary register was addressed, BCTA AS IN L or BCTA AS OUT L inhibits gate E001-pin 13, which inhibits MBSC TRA from being applied to E012-pin 12. The other two inputs to E012-pin 12 are DEMAND (1) H (doing a Massbus cycle) and BCTA CTOD H (indicating a write data transfer) which, when enabled, generate BCTB SET SSYN L which is applied to the direct-set input of the SSYN flip-flop.

**NOTE**

**The Attention Summary register is a 1-bit per drive pseudo register. When this register is accessed, more than one drive may respond. Therefore, the TRA signal cannot be used to indicate the availability of data. In order to ensure that all drives have their respective Attention Summary bits loaded, a 1.5 microsecond delay is incorporated before the setting of SSYN.**

2. Setting SSYN–Reading Remote Register – If the RH70 is *reading* a remote register (accepting data from the drive), the drive, after recognizing its unit select code, the negation of direction of transfer (CTOD H) and DEMAND, issues TRA and the data. TRA, BCTB DEMAND (1) H and BCTA C1 L are applied to one-shot multivibrator E008. BCTB DEMAND (1) H is asserted (denoting a Massbus cycle) and BCTA C1 L is high (denoting a read from a device). With these conditions present, the one-shot is fired. The external components are selected to provide the 225-ns pulse, designated BCTB DESK DATA L. This delay is designed to allow the data from the drive to propagate to the RH70 and onto the Unibus before SSYN is set. In other words, SSYN cannot be asserted on the Unibus until the data from the drive has been transferred to the Unibus and has stabilized. Consequently, when the one-shot fires, the 0 output goes low for 225 ns, forcing the clock input to the SSYN flip-flop low for this period. At the end of 225 ns, the positive-going trailing edge of the pulse sets the SSYN flip-flop (E017) which is asserted on the Unibus as BUS SSYN L.

3. Setting SSYN–Access Local Registers – SSYN is set during the access of local registers as a result of BCTB REG STR H and BCTB INH DEM H being asserted. REG STR H is generated 85 ns after MSYN and is used as a strobe input to the local register being accessed. The signal INH DEM H indicates that the Massbus cycle is inhibited and a lo cal register is being accessed. At the trailing edge of the REG STR signal, SSYN is set, indicating that data has been accepted or is present on the Unibus when writing or reading a local register.

### 4.3.6 One-Shot Multivibrator (1.5 μs)

The 1.5 μs one-shot multivibrator (E00)8 serves two purposes. First, it determines whether a non-existent drive has been accessed. The one-shot is fired when DEMAND is asserted. If the drive does not respond with MBSC TRA within 1.5 μs, the one-shot times out and the positive-going trailing edge at the 0 output clocks the SET NED (Set Non-Existent Drive) flip-flop set if the Attention Summary register is not addressed (BCTB AS REG H).

If the RH70 issues BCTB DEMAND (1) H and the drive responds with MBSC TRA within 1.5 μs, the one-shot is cleared. The MBSC TRA signal generates BCTB DESK DATA L (reading remote register) or BCTB SET SSYN L (writing remote register). Either of these signals direct-clears both the 1.5 μs one-shot and the SET NED flip-flop via gate E013-pin 6.

**NOTE**

**If the drive does not respond within 1.5 microseconds, the BCTB SET SSYN L (writing remote register) and BCTB DESK DATA L (reading remote register) signals are inhibited, preventing the 1.5 microsecond one-shot and the SET NED flip-flop from being cleared. This action causes the one-shot to time-out and the SET NED flip-flop to set.**

The BCTA DEV SEL H signal is applied to the direct-clear input of the one-shot and SET NED flip-flops. If set, the flip-flops are cleared after the Unibus cycle is completed and the addressed register has been deselected.

A second function of the one-shot is to provide a 1.5 μs waiting period to allow the Attention Summary register in the various drives to be properly read from or written into. In this case, the one-shot times out because the TRA signal is inhibited by BCTA AS IN L or by BCTA AS OUT L and at the end of 1.5 μs, SSYN is set. BCTA AS IN L is associated with writing the AS register and BCTB AS OUT L is associated with reading the AS register. The SET NED flip-flop is not set, since it can be set only when a register other than the Attention Summary register has been addressed and no TRA is received from the drive.

### 4.3.7 Data Buffer Out Clock (DB OCLK H)

The DB OCLK H signal is used to release data at the output of the data buffer when the DB register is read by the program. The BCTA CO L signal is used to inhibit the assertion of DB OCLK during Unibus DATIP operations. This is necessary so that a read-modify-write instruction does not falsely remove data from the data buffer. DB register selection is used for maintenance purposes when verifying the operation of the data buffer. DB OCLK is asserted when the RH70 responds with SSYN to the register operation (if not a DATIP) and is released when the processor removes MSYN.

### 4.3.8 SSYN and TRA Light Emitting Diodes (LEDs)

When the SSYN flip-flop in the RH70 or the TRA signal on the Massbus is stuck in the asserted condition, it can hang up the Unibus. The SSYN and TRA LEDs are provided for Field Service and maintenance personnel so they may visually monitor either of these conditions. If the LED is illuminated in this case, it indicates that the associated signal is stuck in the asserted state.

## 4.4 LOGIC DIAGRAM BCTC

This diagram contains the interrupt control logic to prepare the Unibus to do an interrupt. The logic contained herein is similar to that on the M7821 Interrupt Control module, which can be found in the *PDP-11 Peripherals and Interfacing Handbook.*

### 4.4.1 Interrupt Request

The interrupt control logic is initiated by CSTB INTR REQ L which is asserted if: 1) the IE bit and the Special Condition (SC) bit are set while the RH70 is in the RDY state (not doing a data transfer command) or 2) the RH70 has completed a data transfer and the RDY bit has gone from a 0 to a 1 or 3) the program writes 1s into the IE and RDY bit positions of the CS1 register.

This logic is implemented on logic diagram CSTB.

THE CSTB INTR REQ L signal is a request to interrupt the Central Processor and is applied to priority jumper plug E022 which causes a bus request on a specific BR level. Normally for this device, BUS BR5 L is asserted. The other BR outputs from the plug are unasserted at this time. BUS BR5 L causes a bus request on the Unibus. When the processor is ready to allow the RH70 to become bus master, it returns BUS BG5 IN H which is routed through the priority jumper plug and is output as BCTC BG IN H. This signal performs the following functions:

1. It is applied to the clock input of the GRANT flip-flop. However, the data input to GRANT is inhibited by the low output of gate E030-pin 13. The output of this gate is low since BBSY is not set and INTR REQ is asserted which forces the output of E025-pin 3 to go high. This is applied to one input of E030-pin 13. Consider that the other input to E030 is high since BUS NPR L is not asserted at this time.

2. The SACK flip-flop is set 100 ns after the BG IN H signal is issued. The set input to the SACK flip-flop is enabled because the GRANT flip-flop is reset and the BG IN H signal is asserted. This forces both inputs to gate E030-pin 10 low, which causes the output to go high to set the SACK flip-flop. The setting of the SACK flip-flop removes the BUS BR5 L level from the Unibus by inhibiting gate E027-pin 1. BUS SACK L is asserted on the Unibus.

   When the processor receives SACK, it removes the BG IN H signal. At this point, the RH70 waits for SSYN and BBSY from the previous cycle on the Unibus to become unasserted. With BUS BBSY L and BUS SSYN L unasserted and the removal of BG IN H, gate E021-pin 8 is enabled which clocks the BBSY flip-flop set.

NOTE

**When BBSY is set, it enables the set input to the GRANT flip-flop through gates E025-pin 3 and E030-pin 13. The next BG IN H signal from the processor will set the GRANT flip-flop which will pass the BG IN H signal to the next device on the bus.**

Note that the data input to BBSY is high because SACK is set. The setting of BBSY clears the SACK flip-flop via gate E025-pin 3, gate E030-pin 13, and gate E021-pin 12.

In addition, BUS BBSY L and BUS INTR L are asserted on the Unibus along with the 7-bit vector address of the device (the vector address being 224 for the TWU45 subsystem). The vector is jumper-selectable. If the jumper is left in, the corresponding bit is a 1; if the jumper is cut, the corresponding bit is a 0.

### 4.4.2 Interrupt Done

When the processor has accepted the interrupt vector, it asserts BUS SSYN L which is used to generate BCTC INTR DONE via gate E025-pin 11. INTR DONE is used to terminate the interrupt sequence by direct-clearing the IE and the INTR flip-flops on diagram CSTB. This negates the CSTB INTR REQ signal, which in turn, direct-clears the BBSY flip-flop. The clearing of the BBSY flip-flop removes the BUS BBSY, BUS INTR signals and the vector address.

### 4.4.3 BG IN, SACK and BBSY Light-Emitting Diodes (LEDs)

The BCT module contains BG IN, SACK and BBSY LED indicators, which are used by Field Service and maintenance personnel to determine if one of these signals is hanging up the Unibus. For example, if the BBSY flip-flop is stuck in the set state, it will illuminate the associated LED. This will allow Field Service personnel to quickly correct the problem.

## 4.5 LOGIC DIAGRAM BCTD

The logic diagram contains the Unibus data transceivers for driving data on or receiving data from the Unibus. When reading a register, data is driven onto the Unibus from the internal bus (BUSI D00 OUT L - D15 OUT L). This internal bus accepts multiplexed data from the open-collector multiplexers in the RH70 register control logic. The data is enabled by the assertion of BCTB SSYN (1) L. When SSYN is negated, the data is removed from the Unibus.

When writing a register, the Unibus data is supplied to the RH70 as BCTD D00 IN H - D15 IN H. These control lines feed all RH70 registers and also feed the Massbus when writing remote registers.

## 4.6 LOGIC DIAGRAM AWRA

This logic diagram contains the DBL (Double) flip-flop, the CNT DWN (Count-Down) flip-flop, bits 1 - 11 of the BA Bus Address register and the associated control logic. The Bus Address register increments by two for single-word operations or by four for double-word operations. The register also decrements by two for single-word reverse operations or by four for double-word reverse operations.

### 4.6.1 DBL Flip-Flop

The DBL flip-flop determines whether the register counts by two or by four, while the CNT DWN flip-flop determines whether the register counts up or counts down. The DBL flip-flop is clocked by the CSTC CNTLX REQ L signal. The X designates Controller A, B, C or D since this logic print is common to all controllers. The CNTLX REQ signal is asserted when the REQ flip-flop on diagram CSTC is set. This signal is sent to Cache and also clocks the DBL flip-flop set or clear. This flip-flop controls the C lines to Cache, designating whether a single- or double-word transfer is to occur. The DBL flip-flop also controls the single- or double-word operation of the Data Buffer, the Word Count register, and the Bus Address register.

Note that the output of AND-OR gate E058-pin 8 is applied to the DBL flip-flop. When the output of this gate is driven low, the DBL flip-flop is inhibited from setting, and the AWRA EN DBL H signal is negated. In this situation, single-word operation will take place. The conditions under which single-word operation will occur are described below.

**AWRE WC HIB=1 and AWRD WC LOB=1**

When asserted, these signals designate that the word count is all 1s, representing a last word condition, thereby inhibiting double-word operation.

**AWRA CNT DWN (1) H and AWRA BA 01 (0) H**

When asserted, these signals indicate that the Bus Address register is decrementing and is on a single-word boundary which is not a double-word by definition.

NOTE

**When memory addresses are incrementing and BA01 is a 1, the word is on a single-word boundary. Conversely, when memory addresses are decrementing, and BA 01 is a 0, the word is on a single-word boundary (see Paragraph 3.3.2).**

**AWRA CNT DWN (0) H and AWRA BA 01 (1) H**

When asserted, these signals indicate that the Bus Address register is incrementing and is on a single-word boundary, which also prevents double-word operation.

**CSTA BAI (1) L**

When asserted, this signal indicates that the bus address is inhibited from incrementing or decrementing and thus causes only single-word transfers to occur.

**MDPD LAST WORD H and CSTB READ EXCP H**

When a read operation is taking place and a drive error occurs, the RH70 will attempt to continue transferring all remaining words in the data buffer. If the last word in the buffer is part of a double-word data block, the RH70 would normally wait for a second word before accomplishing the memory transfer. In the first word of a double-word operation, the drive will not send the second word,

due to the error condition. However, the RH70 expects to receive this word; in this case, the last word in the data buffer will assert MDPD LAST WORD H, which will inhibit the AWRA EN DBL H signal and force a single-word operation to occur.

The DBL flip-flop is cleared by the AWRE CLR signal or by loading a data transfer command into the drive (GO CLR L).

### 4.6.2 CNT DWN Flip-Flop

The CNT DWN flip-flop determines whether the Bus Address register is to be incremented or decremented. This flip-flop is clocked by the trailing edge of the CSTA GO CLR L signal which is asserted when a data transfer command has been loaded. The function codes for the various commands are listed in Paragraph 3.5. Note that all reverse operations have function code bits F0 and F1 (corresponding to BCTD D01 IN H and BCTD D02 IN H, respectively) asserted. Since the GO bit is also asserted, the least significant octal digit of the command code is a 7. When these conditions are met, the CNT DWN flip-flop is set and will cause the Bus Address register to decrement. The CNT DWN flip-flop is cleared by the CSTD CLR H signal, which is asserted as a result of a program clear, DC LO signal or INIT signal.

### 4.6.3 BA01 Flip-Flop

The BA01 flip-flop represents bus address bit 01. For single-word operations, this flip-flop is toggled, causing the address to be incremented or decremented by two. There is no countup or countdown logic associated with bit 01 due to the following reasons. If the flip-flop is reset, incrementing or decrementing the address will set the flip-flop. Conversely, if the flip-flop is set, incrementing or decrementing the address will reset the flip-flop.

**NOTE**

**For double-word operations, this flip-flop is not toggled; instead bus address bit 02, which is part of a 74193 4-bit binary counter chip, is employed. This bit causes the bus address to increment or decrement by four.**

The BA01 flip-flop is a J-K type with J and K inputs tied together. In this configuration, when a high input is applied to J and K and the clock input is clocked, the flip-flop will toggle. If a low input is applied to the J and K inputs and the clock input is clocked, it has no effect on the flip-flop. Note that the AWRA DBL (0) H signal (DBL flip-flop reset) is applied to the J-K inputs, which means that the flip-flop will toggle each time a clock is applied (single-word operation). If the DBL flip-flop is set, the J and K inputs are driven low and the flip-flop is locked in its present state.

The clock signal to the BA01 flip-flop is asserted if the CDPJ SELADRS CNTLX H signal and the AWRD RACK H signals are asserted and the BAI flip-flop is not set. CDPJ SELADRS CNTLX H is a level from the Cache which indicates that this controller has been selected. The AWRD RACK H signal is a pulse which occurs during the RH70 memory cycle with Cache and indicates that the Cache has acknowledged the request made by this controller and has stored the bus address and control information. The CSTA BAI (0) H is the bus address increment inhibit signal. If this flip-flop is reset, the bus address will increment or decrement in its normal fashion. The BA01 flip-flop is loaded or cleared by the direct-set or direct-clear input. BCTB REG STR H, BCTA LO BYTE H, BCTA BA IN L and the state of BCTD D01 IN H from the Unibus determine whether the flip-flop is loaded or cleared. The first three signals indicate that the low byte of the Bus Address register is being written during the time of the register strobe. If D01 is asserted, the flip-flop is clocked set; if D01 is negated, the flip-flop is clocked reset.

### 4.6.4 BA02, BA03 Bus Address Bits

There are three 74193 4-bit binary counter chips on this diagram which yield bus address bits 02 - 11 (AWRA BA 02 H - AWRA BA 11 H).

Note that bus address bits 02 and 03 are on one chip, bits 4 - 7 are on a second chip, and bits 8 - 11 are on a third chip. The bits are implemented this way since the bus address is loaded as a byte and this configuration minimized the logic required. Note that bits 2 - 7 are loaded by AWRA LOAD BALB L and bits 8 - 11 (and 12 - 15 on AWRB) are loaded by AWRA LOAD BAHB L, corresponding to the low byte and high byte, respectively.

As previously stated, BA02 is incremented or decremented 1) during double-word operations, or 2) when a carry propagates from BA01. During double-word operations, gates E053-pin 6 and E053-pin 3 are enabled. If the bus address is being *incremented*, AWRA CNT DWN (0) H is asserted and enables gate E060-pin 6, which will cause

binary counter chip E043 to increment on the trailing edge of the AWRD RACK H signal. If the bus address is being *decremented*, AWRA CNT DWN (1) H is asserted and enables gate E060-pin 8, which will cause binary counter chip E043 to decrement on the trailing edge of AWRD RACK H. Note when decrementing or incrementing during double-word transfers, that the BA01 bit is inhibited from changing state due to the negation of the AWRA DBL (0) H signal.

In the case where a carry is propagated from bit BA01, assume that the operation is a single-word transfer and the address is being incremented. In this case, BA01 is a 1 which enables gate E053-pin 3. This in turn enables gate E060-pin 6, causing binary counter chip E043 to increment on the trailing edge of AWRD RACK H. In addition, the BA01 flip-flop is toggled, since the DBL input is no longer holding it disabled. Consequently, in this case, BA02 incremented while BA01 went from a 1 to a 0. For decrementing, a similar analysis is followed with BA01 on a 0. Gates E053-pin 6 and E060-pin 8 are implemented. On the trailing edge of AWRD RACK H, BA02 decrements and BA01 changes from a 0 to a 1.

Since the upper two bits of binary counter E043 are not used, the carry and borrow condition for bus address bits 02 and 03 must be calculated. A carry condition is generated if BA01, BA02 and BA03 are asserted, or if BA02 and BA03 are asserted and a double-word operation is to occur (see gate E049-pin 6). The output of this gate enables NAND gate E054-pin 8 to increment BA04 during incrementing [AWRA CNT DWN (0) H], if AWRA COUNT BA H is asserted. COUNT BA H is asserted if this RH70 has been selected, Cache has issued the REQ ACKN signal to the RH70, and the Bus Address Increment Inhibit flip-flop is reset (see AND gate E059-pin 6).

When bus addresses are decremented a borrow condition must be calculated. A borrow is generated if BA01, BA02 and BA03 are all negated or when BA02 and BA03 are negated and a double-word operation is specified. This is implemented in NOR gate E055-pin 8. The output of this gate enables NAND gate E054-pin 6. The other inputs to this gate are enabled when decrementing addresses [AWRA CNT DWN (1) H] if AWRA COUNT BA H is asserted.

#### 4.6.5 BA04 - BA11 Bus Address Bits

Binary counter E047, which is associated with bus address bits 04 - 07, generates an internal borrow or carry to the next counter chip E046 as all bit positions of E047 are used. The count up input to E046 is incremented by a carry while the count down input is decremented by a borrow. Note that AWRA LOAD BALB L (Load Bus Address-Low Byte) is used to load the low byte of the bus address (bit 01 - 07). Bus address bits 08 - 11 are part of the upper byte and is loaded by AWRA LOAD BAHB L (Load Bus Address-High Byte).

### 4.7 LOGIC DIAGRAM AWRB

This diagram contains the upper part (bits 12 - 15) of the high byte of the Bus Address register, contains the Bus Address Extension register (bits 16 - 21), and the control parity logic.

#### 4.7.1 BA12 - BA15 Bus Address Bits

Bus address bits 08 - 11 are shown on logic diagram AWRA; bus address bits 12 - 15 are shown on logic diagram AWRB. Bits 12 - 15 represent the upper bits of the high byte of the bus address register and are loaded by AWRA LOAD BAHB L. The carry and borrow outputs of binary counter E047 on sheet AWRA, which process bus address bits BA07 - BA11, are applied to the Countup (CUP) and Countdown (CDN) inputs of binary counter E041 (sheet AWRB), which process bus address bits BA12 - BA16.

#### 4.7.2 Bus Address Extension Register (Bits BA16 - BA21)

The borrow and carry outputs of binary counter E041 are applied to the CDN and CUP inputs to binary counter E035, which processes bus address extension bits 16 - 19. The borrow and carry outputs of E035 are applied to the CDN and CUP inputs of binary counter E036, which processes bus address bits BA20 and BA21. Consequently, the borrow or carry propagates through the Bus Address register to bits 20 and 21 of the Bus Address Extension register.

If the program is writing bits A16 and A17, it can write these bits in two registers: bits 8 and 9 of the CS1 Control and Status 1 register (high byte) and in bits 00 and 01 of the BAE (Bus Address Extension) register (low byte). Note that there is only one load input into binary counter E036. Consequently, AND-OR gate E045-pin 8 is provided to allow loading from the CS1 register or from the BAE register.

If A16 and A17 are written via the BAE register, the upper AND gate is enabled (AWRB BAE IN H, BCTA LO BYTE H, and BCTB REG STR H). BCTD D00 - D03 in H are loaded via multiplexer E040 into A16 and A17 positions of binary counter E036. If A16 and A17 are written via the CS1 register, the lower AND gate is enabled (BCTB REG STR H, BCTA HI BYTE H, BCTA CS1 IN L and CSTA RDY H).

Since binary counter E035 is a four-bit chip and the program is writing bits A16 and A17, AWRB A18 H and AWRB A19 H are fed back to the input multiplexer (E040) to preserve their contents.

### 4.7.3 Asynchronous Massbus Parity (CPA OUT)

The two 74180 chips shown on sheet AWRB are employed for parity generation when writing into a register in the drive. The 16 data inputs to the chips are from the Unibus data lines. Note that odd parity is normally selected. (CSTA PAT H is normally negated.) Parity on the Massbus is odd.

Assume that the data inputs contain an even number of 1s. This is summed with the ODD input to assert the EODD output which generates PA OUT H. This is the parity bit generation and is supplied to the Massbus driver for transfer to the drive. If the data inputs contain an odd number of 1s, the PA OUT H signal is not asserted and no parity bit is generated. When reading from a drive register, a different set of data lines is used and this necessitates two additional 74180 chips to check parity. (See sheet AWRF.)

The PAT H signal can be asserted by the program (bit 4 in CS2) to generate even parity on the Massbus. This maintenance feature allows verification of the parity logic in the drive.

## 4.8 LOGIC DIAGRAM AWRC

This logic diagram contains the BUSI internal bus multiplexers for the Bus Address and Bus Address Extension registers. The diagram also shows the address drivers which drive bus address and bus address extension information from the RH70 to the MBCBUS, which is the RH70/Cache Interface.

### 4.8.1 BA and BAE Multiplexers

The four open-collector multiplexers (E020, E024, E026 and E030) gate the bus address and the bus address extension bits onto the BUSI internal bus. If BCTA BA OUT L is asserted, the A0 - A3 inputs of each multiplexer are supplied at the output which feed the bus address onto BUSI. If BCTA BAE OUT L is asserted, the B0 - B3 inputs of each multiplexer are supplied to the output.

This gates the bus address extension bits onto BUSI. Since the bus address extension is only six bits, the remaining ten inputs to the B side of the multiplexer are grounded. The BUSI internal bus, in turn, is used to provide the appropriate register contents to the Unibus data lines when the program reads a register.

### 4.8.2 Address Drivers

There are 21 address drivers for the Bus Address and Bus Address Extension Registers that are used to drive address information onto the MBCBUS which serves as the RH70/Cache Interface. The MBCBUS lines are common to all RH70 Controllers. The address is gated onto the MBCBUS by the CDPJ SEL ADRS CNTLX signal for this controller.

**NOTE**

**This diagram is common to all four RH70 Controllers. The X in CNTLX actually refers to the Controller (Controller A, B, C or D) that has been selected by the Cache.**

### 4.8.3 Control Signals CX and C1

There are two drivers for driving the CX and C1 signals onto the MBCBUS. When asserted, the CX signals, specify a double-word operation and when unasserted, specify a single-word operation. CX is not used during a write operation or write-check operation, since double words are normally transferred from memory to the RH70. If a single-word operation has been specified, the memory will still transfer a double word; however, only the appropriate word is clocked into the RH70; the other one is disregarded. During a read operation, the CX line is used to specify writing a double (CX asserted) or single word (CX negated) into memory.

**NOTE**

**The C0 signal (if it was implemented) defines read pause or write byte. However, since the subsystem only reads or writes words, this bit is not used.**

The chart below shows the various control signal combinations for the data transfer operations.

| OPERATION | MBCBUS CX L | MBCBUS C1 L |
|---|---|---|
| Write, write-check | unasserted | unasserted |
| Read (double word) | asserted | asserted |
| Read (single word) | unasserted | asserted |

## 4.9 LOGIC DIAGRAM AWRD

This diagram contains the low bytes of the Drive Word Count register (DRWC) and the Word Count register (WC).

### 4.9.1 Drive Word Count Register

The Drive Word Count register is loaded in parallel with the Word Count register. CSTB SYNC CLK B H is applied to the Countup (CUP) input of the register and causes the register to be clocked on the trailing edge of the SYNC CLK signal. This register is incremented by one, since single words are gated on/off the Massbus.

Bits 00 – 03 of the Drive Word Count register are located on E037; bits 04 – 07 are located on E032. The Carry (CRY) output from E037 is fed to the Countup (CUP) input to E032. The CRY output of E032, in turn, is fed to the high byte of the Drive Word Count register (see logic diagram AWRE), in order to propagate the carry through the register.

The Drive Word Count register is loaded by ANDing BCTA WC IN L, BCTA LO BYTE H and BCTB REG STR H, which is similar to the loading of the other RH70 registers.

### 4.9.2 Word Count Register

The Word Count register for the low byte is configured differently than the Drive Word Count register. Bit 00 of this register is a single J-K flip-flop with J and K inputs connected together. Consequently, a clock signal with asserted J-K inputs cause the flip-flop to toggle, while a clock signal with negated J-K inputs has no effect. For double-word operation, WC00 is prevented from toggling due to AWRA DBL (0) H being unasserted. Instead, WC01 is toggled for each double word. During single-word operation, WC00 is toggled with each clock input applied to the register. Only bits 01 – 03 of the Word Count register are located on E022 since WC00 is implemented as a single flip-flop. This leaves one unused input on this counter since it is a four-input device. It is necessary therefore to calculate the carry from this stage. Bits 04 – 07 are located on E017. Note that the output of WC00 or DBL (0) H is applied to E042-pin 11 which then enables a carry condition to E042-pin 8, E028-pin 8, and E005-pin 6. A carry condition from bits 01 – 03 is applied to E028-pin 8 and E005-pin 6 via gate E005-pin 8. In effect, this configuration ANDs the bits of the low byte to determine if a carry is propagated to the next stage. For example, if bits 00 – 03 are all asserted, or if bits 01 – 03 are asserted and double-word operation is specified, AND gate E028-pin 8 is qualified when the register is clocked, causing bit 04 in E017 to increment. Similarly, if bits 00 – 07 are all asserted, or bits 01 – 07 are asserted and double-word operation is specified, AND gate E005-pin 6 is asserted yielding AWRD WC LB CRY H. This signal is applied to the high byte of the Word Count register to increment the E011 low order stage in that register (see logic diagram AWRE).

**NOTE**

**The MAX output on pin 12 in E017 is asserted if the counter is incremented to all 1s. The MIN output is not utilized as the word counter is only allowed to increment and not to decrement.**

The Word Count register is clocked in the same manner as the Drive Word Count register. When the bus address is incremented, the word count is incremented. This occurs as a result of the CCBE REQ ACKN L signal from the Cache and the CDPJ SEL ADRS CNTLX H signal which specifies this controller. These signals are ANDed in AND gate E005-pin 12 to yield AWRD REQ ACKN H. Note that the entire Word Count register is clocked by AWRD REQ ACKN H (see logic diagram AWRE).

### 4.9.3 Detection of Maximum Word Count

When bits 00 - 07 of the Word Count register are all 1s, AWRD WC LOB 1 H is asserted at the output of E003-pin 10. This signal is ANDed with AWRE WC HIB 1 H on logic diagram AWRE to detect an all 1s condition in the Word Count register. This all 1s condition signifies that the last word is being transferred and inhibits the data input to the DBL flip-flop (see logic diagram AWRA).

## 4.10 LOGIC DIAGRAM AWRE

This diagram contains the high bytes of the Drive Word Count and the Word Count registers. They are incremented similar to the drive word count low byte and word count low byte, described on logic diagram AWRD. Since all chips on this diagram are four-input chips, with all inputs being used, the carry from one chip is rippled to the Count-up (CUP) input of the next chip.

The high bytes of the Drive Word Count and Word Count registers are loaded similarly to the low bytes previously described. Note, however, that BCTA HI BYTE H is used instead of BCTA LO BYTE H.

### 4.10.1 Drive Word Count Overflow

For Drive Word Count, when all bits are 1s and a carry is propagated through the register, the low-going carry from the most significant bit (bit 15) is inverted and is applied to the clock input of the Drive Word Count Overflow (DRWC OFLO) flip-flop as a positive-going transition. Consequently, this flip-flop will be clocked on the leading edge of the clock pulse, and causes AWRE DRWC OFLO (0) H to be asserted. This signal inhibits further Massbus cycles by causing the CSTB DIS SCLK (Disable Sync Clocks) flip-flop to set.

The Drive Word Count overflow flip-flop is cleared by the CSTA CLR + GO CLR L signal,which is asserted by CD LO, INIT, PROGRAM CLEAR or by initializing the RH70 during a data transfer command.

### 4.10.2 Word Count Overflow

For Word Count, when all bits are 1s and a carry is propagated through the register, the carry (RIP CLK) output of WC15 is applied to the WC OFLO flip-flop, which causes this flip-flop to set on the trailing edge of the clock pulse. This asserts AWRE WC OFLO (0) H which inhibits the CSTC SMC (Start Memory Cycle) flip-flop from setting, thus terminating further memory data transfers.

The Word Count Overflow flip-flop is cleared by CSTA CLR + GO CLR L, which is asserted by DC LO, INIT, PROGRAM CLEAR or by initializing the RH70 during a data transfer command.

## 4.11 LOGIC DIAGRAM AWRF

This logic diagram contains the BUSI open-collector multiplexers, the parity checkers, and the Massbus Control Parity Error (MCPE) flip-flop.

### 4.11.1 BUSI Multiplexers

The BUSI open-collector multiplexers (E007, E008, E013 and E014) gate the word count or the contents of a remote register onto BUSI. BUSI is the internal bus which is driven onto the Unibus data lines. When reading the Word Count register, the multiplexer gates the word count (AWRD WC 00–WC 07 and AWRE WC08–WC15) onto the Unibus data lines. When reading a Remote Register, the multiplexer gates the Massbus C lines (C00 IN H - C15 H) onto the Unibus data lines. The word count is enabled onto BUSI via BCTA WC OUT L and the Massbus C lines are enabled onto BUSI via BCTA CNTL OUT L.

### 4.11.2 Parity Checker, MCPE Flip-Flop

The parity checkers monitor the CPA parity bit and the 16 Massbus control bus data lines when a remote register is being read. If the parity is odd, the EVEN output is forced low and inhibits the Massbus Control Parity Error flip-flop from setting. However, if a parity error occurs (even parity), the EVEN output is asserted and enables the set input to the MCPE flip-flop. When the flip-flop is clocked by the negation of BCTA CNTL OUT L signal (asserted during a remote register read operation), the MCPE flip-flop is set. This condition is reflected in bit 13 of the CS1 register. When the MCPE flip-flop is set, the reset output latches the flip-flop set via NAND gate E010-pin 6.

When reading the Attention Summary (AS) register, the parity check logic is inhibited, since it is not known how many devices will respond with their respective Attention Summary bits. Consequently, when the AS register is read, BCTB AS REG H is asserted which inhibits the setting of the MCPE flip-flop.

The MCPE flip-flop is cleared by CSTB CLR ERR L which is a function of CLR . GO CLR, or the posting of a 1 in the TRE bit position (bit 14) of the CS1 register.

## 4.12 LOGIC DIAGRAM MDPA

This sheet contains the control logic which controls the clocking sequence for the RA, RB, RC, and RD registers. Refer to Figures 4-5, 4-7, and 4-8 for the following detailed description of the data buffer.

### 4.12.1 RA FULL and RB FULL Flip-Flops

The RA FULL and RB FULL flip-flops are shown in zone D2 and C2 and indicate the status of the RA and RB registers, respectively. MDPA CLK RA H is the signal used to clock the RA FULL flip-flop; MDPA CLK RB H is the signal used to clock the RB FULL flip-flop. These flip-flops are clocked on the leading edge of the clock signals. When either flip-flop is clocked, the set output goes high, indicating that the respective register is full.

The RA FULL flip-flop is cleared by the general-purpose initialize pulse (CSTD DB INIT L) or by the MDPA CLR RA L signal. MDPA CLR RA L is generated approximately 50 ns after the data has been clocked from RA into RC. (Note that MDPA CLK RC H occurs at the 100-ns tap on delay line E102 and at the 150-ns tap, AND gate E100-pin 6 is asserted low which creates MDPA CLR RA L to direct-clear the RA FULL flip-flop.) The 100-ns delay ensures that the data has been transferred to RC and has had time to stabilize.

The RB FULL flip-flop is cleared in exactly the same manner as the RA FULL flip-flop. In other words, approximately 50 nanoseconds after the RD register is clocked, MDPA CLR RB L is generated (note the 150-ns tap on delay line E091). This ensures that the data has had time to be transferred from RB to RD and stabilize. At this point, the RB FULL flip-flop can be cleared.

### 4.12.2 MDPA CLK RA H, MDPA CLK RBH

MDPA CLK RA H is derived from AND-OR gate E113-pin 8 as a result of one of the following conditions:

1. In a read operation, when the contents of OBUF is being clocked into RA (two-input AND gate–pins 2, 3)

2. In data buffer maintenance operation, when Unibus data is being clocked into RA (three-input AND gate–pins 4, 5, 6)

3. When memory data is being clocked into RA or RB, or both, during a Write or Write-Check command (four-input AND gate 1, 11, 12, and 13).

MDPA CLK RB H is derived from AND-OR gate E104-pin 8 as a result of one of the following conditions:

1. In a Read operation, when the contents of OBUF is being clocked into RB during a Read command (two-input AND gate–pins 2, 3)

2. In data buffer maintenance operation, when Unibus data is being clocked into RB (three-input AND gate–pins 4, 5, 6)

3. When memory data is being clocked into RA or RB, or both, during a Write or Write-Check command (four-input AND gate–pins 1, 11, 12, 13).

**4.12.2.1 Assertion of CLK RA, CLK RB During Read Command** – The topmost AND gate in E113-pin 8 is grounded and is not used. In a Read command, data from the drive is transferred to RE, sequences up to OBUF, and is transferred to RA or RB, then to RC or RD for transfer to memory. The two-input gate in E113-pin 8 is asserted during a Read command if OBUF is full, Input Ready (IR) is asserted, and the RB ENA pointer is pointing to an even word. Assertion of the IR signal (MDPA IR H) indicates that the RA register is empty and has been selected to receive a data word from OBUF or that the RB register is empty and the RB register is selected to receive a word from OBUF. MDPA IR H, MDPH READ H, and MDPD OBUF FULL (1) H are ANDed in AND gate E114-pin 8. The output of this gate is driven down delay line E119 by driver Q6. 100 ns after the start of the delay line, AND gate E114-pin 12 is qualified. If the RB ENA pointer is selecting the even word [MDPD RB ENA (0) H], the two-input AND gate of AND-OR gate E113-pin 8 is qualified. This causes the MDPA CLK RA signal to be generated which clocks the RA register with the contents of OBUF. If the RB ENA pointer is selecting the odd word [MDPD RB ENA (1) H], the two-input AND gate in AND-OR gate E104-pin 8 is qualified. This causes the MDPA CLK RB H signal to be generated, which clocks the RB register

with the contents of OBUF. At the same time, AND gate E114-pin 6 is asserted, which generates MDPA CLR OBUF H. 50 ns later (150-ns tap on delay line E119), the CLR OBUF signal is negated, and the OBUF FULL flag is cleared, releasing that data word from the OBUF register.

NOTE

**At the 200-ns tap on delay line E119, AND gate E114-pin 12 becomes negated. This means that the CLK RA or CLK RB signal is approximately 100 ns wide, plus additional gate delays.**

The output of AND-OR gate E113-pin 8, used to yield the CLK RA signal, and the output of AND-OR gate E104-pin 8, used to yield the CLK RB signal, are ORed in OR gate E099-pin 3 to assert MDPA INC RB ENA H. On the trailing edge of this signal, the RB ENA pointer is incremented to switch to the opposite side of the data path (odd word to even word or even word to odd word).

**4.12.2.2 Assertion of CLK RA or CLK RB During Maintenance Operation** – When writing the data buffer during maintenance operation, data from the Unibus can be sequenced through the data buffer and transferred back to the Unibus. This is accomplished as described below. BCTA DB IN L from the register decoder enables one input of the three-input AND gate in E113-pin 8 and in one input in AND gate E104-pin 8. A second input to each of these gates is the register strobe signal (BCTB REG STR H) which allows the registers to be clocked with data. A third input to each gate is the RB ENA pointer which determines whether data will be clocked into the RA (even word) or RB (odd word) side of the data path. For example, assume that the data buffer is initialized to point to the RA register [RB ENA (0) H asserted]. The first word written by the program is clocked into the RA register since the CLK RA H signal will be asserted. In addition to clocking the RA register, this signal is applied to OR gate E099-pin 1, generating MDPA INC RB ENA H. This switches the RB ENA pointer, which will now point to the odd word (RB register) portion of the data path. As a result, the next data word from the Unibus will be clocked into the RB register.

**4.12.2.3 Assertion of CLK RA or CLK RB During Write or Write-Check Command** – In a Write or Write-Check command, data from memory is applied to RA and RB. This is accomplished via the four-input AND gate of E113-pin 8 for the CLK RA signal and the four-input AND gate of E104-pin 8 for the CLK RB signal. The inputs to these gates are described below.

1. The CSTA WR + WR CHK H signal designates that a Write or Write-Check command has been loaded into the drive.

2. The CDPK DATA RDY CNTLX H signal is the clock pulse from the RH70/Cache Interface, which is used to clock memory data into the RA or RB register. When this signal is asserted to the RH70 Controller, it means that the Cache arbitration logic has recognized this controller. The X designates one of four RH70 Controller (Controller A, B, C, or D) which may be installed in the system.

3. AWRA DBL (1) L and the sense of the RB ENA pointer. If AWRA DBL (1) L is asserted, it indicates that the DBL flip-flop is set and the bottom gate of E113-pin 8 and E104-pin 8 are both enabled. As a result, both RA and RB will be clocked simultaneously. When the DBL signal is not asserted, the CLK RA or CLK RB signal is generated as a result of the sense of the RB ENA pointer. If MDPD RB ENA (0) H is asserted, RA CLK is generated, which will transfer the even word from memory to the RA register. If MDPD RB ENA (1) H is asserted, RB CLK is generated, which will transfer the odd word from memory to the RB register.

**4.12.3 RC FULL and RD FULL Flip-Flops**

The RC FULL and RD FULL flip-flops are shown in zones B2 and A2 and designate the states of the RC and RD registers, respectively. MDPA CLK RC H is the signal used to clock the RC FULL register; MDPA CLK RD H is the signal used to clock the RD FULL register. These flip-flops are clocked on the leading edge of the clock signals.

The RC FULL and RD FULL flip-flops can be cleared in one of the following ways:

1. By the general-purpose CSTD DB INIT L signal which initializes the data buffer.

2. By the MDPE CLR RC L and MDPE CLR RD L signals, which are asserted after the contents of the RC or RD has been transferred through the MXR to the RE register during write, write-check or data buffer maintenance operations. This occurs approximately 150 ns after the transfer of data from RC or RD into RE.

3. By the CSTC READ DONE L signal, which is a timing signal derived from the RH70/Cache Interface during read operations. When asserted, this signal indicates that memory has accepted the data and the RC or RD register may be cleared in order to allow new data to be loaded into these registers.

### 4.12.4 MDPA CLK RC H, MDPA CLK RD H

Data is clocked into the RC and RD registers under the following conditions:

1. During maintenance operation, or when the program is executing a Write or Write-Check command, data is transferred from RA to RC and/or RB to RD.

2. Normally, during a Read command, double words are transferred to memory from RC and RD. The act of transferring either RA into RC or RB into RD initiates a Cache memory request. Thus, during double-word transfers the RH70 logic forces the assembly of a double word into RA and RB before attempting a transfer to RC and RD.

MDPA CLK RC H is derived from delay line E102 as a result of one of the following conditions: MDPA CLK RD H is generated in a similar manner to delay line E091. The differences between the two are noted.

1. During data buffer maintenance operation, when RA is full [MDPA RA FULL (1) H] and RC is empty [MDPA RC FULL (0) H]. This is implemented in AND gate E107-pin 6.

**NOTE**

**The third input [RD ENA (0) H] to AND gate E107-pin 6 is not required, but is merely implemented to verify that the pointer is switching as required.**

2. During a write or write-check operation, when both RC and RD are empty and RA is full (see AND gate E107-pin 8).

3. During a single-word transfer to memory (read operation), either the three-input AND gate in AND/OR gate E106-pin 8, or the three-input AND gate in AND/OR gate E105-pin 8 is asserted. If the RD ENA pointer is pointing to RC, MDPD RD ENA (1) H is negated and the three-input AND gate associated with E106-pin 8 is qualified, causing the next data word to be transferred from RA to RC. Conversely, if the RD ENA pointer is pointing to RD, MDPD RD ENA (0) H is negated and the three-input AND gate in E105-pin 8 is qualified, causing the next data word to be transferred from RB to RD.

4. During a double-word transfer to memory (read operation), when RA and RB are both full, RC and RD are both empty, a Read command and double-word operation is specified, all inputs to the four-input AND gates in AND-OR gates E106-pin 8 and E105-pin 8 are qualified, which ultimately cause both RC and RD CLK signals to be asserted.

When the output of E106-pin 8 goes low as a result of one of the conditions just mentioned, it is driven down delay line E102 by driver Q6. 100 ns after the pulse is sent to the delay line, AND gate E090-pin 12 is asserted, which causes MDPA CLK RC H to be asserted. Similarly, if E105-pin 8 goes low, this pulse is driven down delay line E091 via driver Q7. 100 ns after this pulse is applied to the delay line, AND gate E090-pin 8 is qualified, which causes MDPA CLK RD H to be asserted. Both signals are cleared 50 ns later (see 150-ns tap on delay line E102, which inhibits E090-pin 12 and the 150-ns tap on delay line E091, which inhibits E090-pin 8).

### 4.12.5 Parity Check Enable

When a write or write-check operation is specified, one input to AND gate E115-pin 10 and one input to AND gate E115-pin 13 is enabled. If the output of E106-pin 8 goes low, MDPA EN EWPCK H is asserted to enable the even-word parity check circuitry, which monitors the contents of the RA register, plus the parity bits. If E105-pin 8 goes low, MDPA EN OWPCK H is asserted to enable the odd word parity check circuitry, which monitors the contents of the RB register, plus the parity bits. When double-word operation is specified, both the odd and even-word parity check circuitry are enabled. For read or data buffer maintenance operations, the parity check enable signals (MDPA EN EWPCK H, MDPA EN OWPCK H) are inhibited.

### 4.12.6 START MEM Enable

When either E106-pin 8 or E105-pin 8 goes low (causing the RC CLK or RD CLK signals, respectively), MDPA START MEM H, at the output of OR gate E100-pin 11, is asserted. This signal is sent to the memory control where error conditions are examined (refer to logic description of logic diagram CSTC).

NOTE

**During data buffer maintenance generations, this signal is inhibited in the memory control logic since no memory transfer takes place in this mode.**

If no errors are present by the time of the REQ CLK signal (which occurs approximately 100 ns later), the next memory request is issued. fifty ns after the data has been clocked into RC, or RD, or both, MDPA CLR RA L, or MDPA CLR RB L, or both, are asserted, which direct-clears the RA FULL or RB FULL flip-flops. At the same time (see the 150-ns tap on delay lines E102 and E091), MDPA CLR SMC H is asserted, which clears the Start Memory flip-flop on CSTC that previously monitored and synchronized the error conditions.

## 4.13 LOGIC DIAGRAM MDPB

This diagram contains the even word memory data registers. This includes the AMX, the RA and RC registers and the MXR logic.

### 4.13.1 AMX Multiplexer

The AMX consists of four and one-half two-input quad multiplexer chips. This multiplexer accepts memory data during write or write-check operations, OBUF data during read operations, and Unibus data during maintenance operations. The OBUF data and Unibus data are multiplexed in the IMX multiplexer shown on logic diagram MDPH. The output of the IMX in turn feeds the AMX. During write or write-check operations, the A inputs (A0, A1, A2, and A3) inputs to AMX are enabled; during read or maintenance operations, the B inputs (B0, B1, B2 and B3) inputs are enabled.

### 4.13.2 Parity Bits (AMX)

A section of the AMX monitors the memory parity bits during write or write-check operations. CDPC MEM BYTE 1 PAR H represents parity for the high byte of the even data word; CDPC MEM BYTE 0 PAR H represents parity for the low byte of the even data word. In the case of data buffer maintenance operations or read operations, the parity signals are unasserted into the RA register.

### 4.13.3 RA, RC Registers

The output of the AMX is applied to the RA register, due to assertion of the MDPA CLK RA H signal. The output of the RA register in turn is applied to the RC register (due to assertion of the MDPA CLK RC H signal) and also to the parity generator/checker logic on logic diagram MDPC.

### 4.13.4 MXR

The output of the RC and RD registers are connected to the Mixer (MXR). If MDPD MXR SEL (1) H is asserted during write, write-check or data buffer maintenance operations, the output of the RD register is applied to the MXR and if MDPD MXR SEL (1) H is not asserted, the output of the RC register is applied to the MXR.

During a read operation, the contents of the RC register is applied to drivers on the RH70/Cache Interface, which eventually asserts data to the even word in memory.

### 4.13.5 Parity Bits (MXR)

Parity bits which are transferred through the MXR are applied to an MXR multiplexer. In this multiplexer, the STB (Strobe) input is used. Consequently, during write operations, this multiplexer forces 0s as parity bits. This prevents false parity from being sent through the data buffer so that correct parity can be generated in the parity generator

between RG and OBUF for transfer to the drive. If a write-check or maintenance operation is specified, the RC or RD parity bits are selected, depending on the state of the MXR SEL flip-flop. If MDPD MXR SEL (1) H is asserted, the parity bits associated with RD are applied through the MXR; if MDPD MXR SEL (1) H is not asserted, the parity bits associated with RC are applied through the MXR.

### 4.14 LOGIC DIAGRAM MDPC

This diagram contains the odd word memory data registers, BMX, RB and RD registers and parity generator/checker circuits. The odd word memory data registers are similar to the even word memory data registers described on logic diagram MDPB. Note that the left-hand side of the diagram shows the BMX which is similar to the AMX on MDPB. However, the BMX is associated with the odd word (that is, bits 16 - 31 from memory). The BMX accepts data from memory during write or write-check operations or accepts data from the IMX multiplexer during read or data buffer maintenance operations. The IMX selects data from OBUF during read operations or selects Unibus data if data buffer maintenance operation is employed.

#### 4.14.1 Parity Bits (BMX)

A section of the BMX monitors the parity bits from memory during write or write-check operations. CDPD MEM BYTE 2 PAR is the parity bit associated with the low byte of the odd word; CDPD MEM BYTE 3 PAR is the parity bit associated with the high byte of the odd word. In the case of data buffer maintenance operations or read operations, these parity signals are unasserted to the RB register.

#### 4.14.2 RB and RD Registers

The output of the BMX is applied to the RB register due to assertion of the MDPA CLK RB H signal. The output of the RB register in turn is applied to the RD register (when the MDPA CLK RD H signal is asserted) and also to the parity generator/checker circuits (shown on this diagram).

#### 4.14.3 Parity Generator/Checker Circuits

The parity generator/checker circuits consist of four chips: parity odd word–high byte, parity odd word–low byte, parity even word–high byte and parity even word–low byte. During read operations, parity is generated by these circuits when data is written into memory. While in write and write-check operations, parity is checked by these circuits. For data buffer maintenance operations, parity is generated as data is written through the data buffer.

The upper two parity generator/checker chips (parity odd word–high byte and parity odd word–low byte) accept the 16-bit data word from the RB register while the two lower parity chips accept the 16-bit data word from the RA register. In addition, each chip has a ninth input, which is the actual parity bit from memory in write or write-check operations and which are 0s for data buffer maintenance operations or read operations. The parity bit for each chip is exclusively-ORed in the XOR gate E103 with one of four invert signals: CSTA INV 0, CSTA INV 1, CSTA INV 2 or CSTA INV 3. The assertion or negation of these signals is controlled by four corresponding control bits in the CS3 register, and allows bad parity to be simulated in order to check the parity generator/checker circuits themselves. During a read operation, the outputs of the parity generator/checker circuits are applied to the RC and RD registers. During write or write-check operations, the parity outputs are applied to logic to see if there is a parity error.

### 4.15 LOGIC DIAGRAM MDPD

This diagram contains the control logic which controls the clocking sequence for the RF, RG and OBUF registers. The diagram also shows the RB ENA pointer, RD ENA pointer, and the MXR SEL pointer.

#### 4.15.1 Clocking the RF Register, RF FULL Flip-Flop

The RF register is clocked on the leading edge of the MDPD RF FULL (1) H signal. Note that this signal is derived from a flip-flop configured as a latch; there is no CLK RF signal as with the other registers. Although the configuration is different, the principle of operation remains the same.

When MDPE RE FULL (1) H (designates RE register full) and MDPD RF FULL (0) H (RF register empty) are asserted, the output of AND gate E060-pin 6 is asserted and is driven down delay line E071 via driver Q4. 100 ns after this signal is applied to the delay line, the set condition of the latch is asserted, which yields MDPD RF FULL (1) H, causing the data from the RE register to be clocked into the RF register.

NOTE

**The TEST INPUT A signal of pin 3 of E060 is a test input for module checkout.**

### 4.15.2 Clearing the RE FULL Flip-Flop

At 150 ns on the delay line, MDPD CLR RE L is asserted, which allows the RE FULL flip-flop to now be cleared since the contents of RE has already been transferred to RF. At 175 ns on the delay line, MDPD CLR RE L is negated, resulting in a 25-ns pulse.

### 4.15.3 Clocking the RG Register, RG FULL Flip-Flop

The data in the RF register is transferred to the RG register upon assertion of MDPD CLK RG H. This signal is dependent on the RG register being empty and the RF register being full. These conditions cause AND gate E060-pin 8 to be asserted. The output is driven down delay line E052 via driver Q3. After 75 ns, MDPD CLK RG H is asserted via AND gate E050-pin 11. The CLK RG signal clocks the contents of the RF register into the RG register, and causes the RG FULL flip-flop to be direct-set.

### 4.15.4 Clearing RF FULL Flip-Flop

50 ns after the CLK RG signal is asserted (125-ns tap on delay line E052), MDPD CLR RF L is asserted, which clears the RF FULL flip-flop to allow new data to be entered into the RF register. 50 ns after the CLR RF signal is asserted (175-ns tap on delay line E052), the signal is negated, resulting in a 50-ns pulse.

The RF FULL flip-flop is cleared by the general-purpose CSTD DB INIT L signal, or approximately 125 ns after the transfer of data from RF into RG by the MDPD CLR RF L signal.

### 4.15.5 RG RDY Flip-Flop

When the RG RDY flip-flop is set, it indicates that the data in the RG register has previously been loaded, the data has had time to settle and stabilize, and the parity has been checked. This is used for Data Late (DLT) or data overrun conditions. Normally, when writing the drive, the data in OBUF is strobed into the drive on the leading edge of SCLK and the data in OBUF is changed on the trailing edge of SCLK. Before the data can be changed however, it must be ensured that a data word is in RG and has stabilized because this is the next word to be transferred to OBUF. The setting of the RG RDY flip-flop indicates that this condition has been satisfied.

The RG RDY flip-flop is set 150 ns after the MDPD RG FULL (1) H signal is asserted. The 150 -ns delay is comprised of two 75-ns delays. The first 75-ns delay is a result of the delay line (150 -ns tap, minus 75-ns tap); the second 75-ns delay is due to the fact that the trailing edge of the 75-ns pulse traveling down the delay line clocks the RG RDY flip-flop. The pulse traveling down the delay line is 75 ns in length since the delay line is turned off at the 75-ns tap by the setting of RG FULL.

The clearing of the RG FULL flip-flop after the data has been transferred to OBUF causes the clearing of the RG RDY flip-flop, which indicates that the data in RG is not valid.

### 4.15.6 Clocking OBUF, OBUF FULL Flip-Flop

To clock a data word from RG into OBUF requires the RG register to be full and the OBUF register to be empty. When these conditions are satisfied, AND gate E060-pin 12 is qualified and the output is driven down delay line E041 via driver Q2. At 150 ns, MDPD CLK OBUF H is asserted via gate E040-pin 8 and gate E062-pin 3. The 150-ns delay allows the data in RG to settle and stabilize and provides additional time for parity checking, since the parity check is situated between RG and OBUF. MDPD CLK OBUF H clocks the data from the RG register into the OBUF register and also direct-sets the OBUF FULL flip-flop, which then turns the delay line off.

During a write operation, once OBUF is full and the OBUF FULL flip-flop is set, OBUF FULL remains set for the entire transfer until an error or word count overflow occurs. It may be recalled in the other data buffer registers that the delay line was used to generate the clock and clear signals necessary to transfer data from one register to the next. However, the delay line is not used in a write operation after the first data word reaches OBUF.

NOTE

**Delay line E041 is used in the normal fashion for read, write-check and data buffer maintenance operations, in addition to the first data word in a write operation.**

At this point, a word is in OBUF and the OBUF FULL flip-flop is set. Assume that successive data words have filled the rest of the data buffer. The RH70 then asserts the RUN line to the drive and the drive proceeds to rotate to the correct address. When the correct address is found, the drive issues a SCLK pulse. The leading edge of SCLK causes the data in OBUF to be transferred to the drive. The trailing edge of SCLK initiates a Drive Clock pulse which causes the data in RG to sequence to OBUF (provided RG RDY is set, indicating that a word has been loaded in RG and has stabilized). Successive SCLK signals cause the data in OBUF to be transferred to the drive, while successive Drive Clock (trailing edge of SCLK) signals cause the data in RG to sequence to OBUF by causing E50-pin 3 and E62-pin 3 to assert CLK OBUF H signals without using the delay line. The process continues until an error condition is flagged or word count overflow occurs.

This process is implemented by the logic as described in Figure 4.8.

1. SCLK (leading edge) – clocks first data word from OBUF to drive.

2. SCLK (trailing edge creates Drive Clock) – second data word in RG sequences to OBUF. This is accomplished by MDPD WRITE H and CSTB DRIVE CLK H, which are ANDed in E050-pin 3 to assert MDPD CLK OBUF H. RG register is now transferred to the OBUF register.

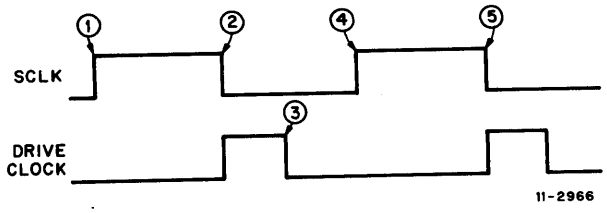



Figure 4-8 Massbus Data Transfer Sequence

3. Drive Clock (trailing edge) – clears RG FULL flip-flop since RG is now empty. This flip-flop is cleared on the positive-going transition from gate E050-pin 3. This gate went negative on leading edge of Drive Clock and goes positive on trailing edge.

   NOTE

   **If RG FULL is reset, the output from the 1 side direct-clears the RG RDY flip-flop, indicating that the data in RG is no longer valid.**

4. SCLK (leading edge) – second data word transferred from OBUF to drive.

5. SCLK (trailing edge) – third data word had transferred from RF to RG 150 ns after RG was emptied (in step 3). This data word now transfers from RG to OBUF, provided RG RDY flip-flop is set, indicating that the data in RG has had time to settle and have the parity checked.

   The transfer of the data word from RG to OBUF is caused by MDPD CLK OBUF H which is asserted as a result of MDPD WRITE H and CSTB DRIVE CLK H.

This process continues until error condition is flagged or word count overflow occurs.

### 4.15.7 Clearing the RG FULL Flip-Flop

The RG register and the RG FULL flip-flop is cleared by MDPD CLR RG L. This signal is generated 50 ns after the data word is transferred from RG to OBUF. (At 150 ns on the delay line, data is clocked from RG to OBUF and at 200 ns, the RG FULL flip-flop is reset by the MDPD CLR RG L signal).

In addition to resetting the RG FULL flip-flop via the delay line, there are two other conditions which can reset the RG FULL flip-flop. One is the general-purpose CSTD DB INIT L signal which intializes the RH70. The second is the positive-going transition from gate E050-pin 3 which clocks the flip-flop reset. This occurs on the trailing edge of Drive Clock during a write operation.

### 4.15.8 Clearing OBUF FULL Flip-Flop

The OBUF register is cleared as a result of SCLK from the drive which clocks the data out of OBUF. The OBUF FULL flip-flop is cleared in one of the following ways:

1. By the general-purpose CSTD DB INIT L signal which initializes the RH70. (This is a direct clear.)

2. On the trailing edge of MDPE CLK MXR WORD H during a write-check operation. On the leading edge of MDPE CLK MXR WORD H, data from the RC or RD register is compared to OBUF data. If the write-check showed no error, the data in the RC or RD register and data in OBUF can be invalidated by clearing the respective Register Full flip-flops.

   If a write-check error (MDPE WCE L) occurs, the data is frozen in OBUF. This keeps AND-OR gate E061-pin 8 low to prevent the clearing of the OBUF FULL flip-flop. When the programmer recognizes the error, it will clear the WCE bit (bit 14, CS2 register), which in turn causes the OBUF FULL flip-flop to clear.

3. When the last data word has been transferred during a write operation (from RG to OBUF). This occurs when there is no word in RG to transfer to OBUF [MDPD RG FULL (0) H]. On the leading edge of Drive CLK, the output of E061-pin 8 goes low. On the trailing edge of CSTB DRIVE CLK H, the positive-going transition from this gate clocks the OBUF FULL flip-flop clear. Consequently, OBUF FULL is cleared on the last transition of Drive CLK during the normal completion of a Write command.

4. During a read operation when the data in OBUF has been clocked into the RA or RB registers. The MDPA CLR OBUF H signal is asserted at the time that the data is clocked into RA or RB. 50 ns later, the negation of CLR OBUF H causes the OBUF FULL flip-flop to clear.

5. During maintenance operation, the OBUF FULL flip-flop is cleared when the program has read the data buffer. This is the result of ANDing BCTB DB OCLK H and BCTA DB OUT L in gate E059-pin 11. The BCTB DB OCLK H signal is asserted when the processor has recognized that the data has been read from the data buffer. On the trailing edge of the BCTB DB OCLK H signal, the OBUF FULL flip-flop is clocked clear.

### 4.15.9 MDPD DB EMPTY L, MDPD LAST WORD H

These signals are used during a read operation when an error has been detected in the drive but not in the controller. In this case, the data in the data buffer is transferred to memory, which allows error correction on those drives using error correction hardware. The error condition from the drive causes the transfer to stop at the next EBL pulse. The data buffer is empty (MDPD DB EMPTY L) when all the register flags are unasserted. These include the RA, RB, RC, RD, RE, RF, RG and OBUF registers which are ANDed in gate E049-pin 8.

In the case where word count overflow has not yet occurred during a read operation and an error has occurred in the drive and not in the RH70. The RH70 contains logic to determine if the operation is a single- or double-word transfer. If the last transfer should be a double word, the RH70 tries to assemble a double word and if an error occurred, preventing the drive from sending additional words, the RH70 would be left waiting for a word which it will never receive. To avert this situation, the RH70 must detect the fact that the last word is in the data buffer and also must force the memory operation to be performed as a single-word rather than as a double-word operation. This is the function of the MDPD LAST WORD H signal. Note that the single-word section of the data buffer must be

empty (RE, RF, RG, and OBUF). In the double-word section, if either MDPA RA FULL H or MDPA RB FULL H is asserted, but not both, MDPD LAST WORD H is asserted. If both RA FULL and RB FULL are asserted, meaning there are two words in the buffer, MDPD LAST WORD H is inhibited, and the operation is performed as a double-word transfer. MDPD LAST WORD H is ANDed with CSTB READ EXC H on logic diagram AWRA. With MDPD LAST WORD H and CSTB READ EXCP H both asserted, the DBL flip-flop is prevented from setting (see logic diagram AWRA), thus forcing a single-word memory operation.

### 4.15.10 Mixer Select (MDPD MXR SEL)

The MXR SEL flip-flop switches the select lines on the mixer multiplexer to alternately point to the RC and RD registers. In write or data buffer maintenance operations, the MXR output is applied to RE. In write-check operations, the contents of the MXR loaded from RC or RD is compared with the contents of OBUF in the Exclusive-OR gates. When the RH70 is initialized by CSTD DB INIT L, the MXR SEL flip-flop is initially set or cleared to control the MXR. If DB INIT is a result of Unibus INIT, PG CLR, or Power Fail, the Bus Address register is cleared [BA 01 (0) H asserted] and the MXR SEL flip-flop is cleared by gate E080-pin 8. When a data transfer command with the GO bit is loaded in to the drive, CSTA GO CLR L is asserted in the RH70, causing a DB INIT.

NOTE

**CSTA GO CLR L is one of the means of generating CSTD DB INIT L. However, the GO CLR signal will not cause the Bus Address registers to clear [does not affect BA 01 (0) H].**

If AWRA BA 01 (1) H is asserted (odd-word boundary), the MXR SEL flip-flop is directly-set, causing the MXR to point to the RD register (even word). If the bus address bit (BA 01) is reset, (even word) the MXR SEL flip-flop remains reset due to the DB INIT signal leaving the flip-flop pointing to the RC register.

The MXR SEL flip-flop is wired to toggle on every clock input. The clock input is MDPE INCR MXR SEL H. On the trailing edge of this signal, the flip-flop is toggled to the alternate state. There is one condition in which the MXR SEL flip-flop is prevented from toggling. This occurs during a data transfer command when the BAI (Bus Address Increment inhibit) bit is set. In this situation, it is desired to do transfers to or from the same location and the same memory data register (either RC or RD).

NOTE

**If the RH70 is in the RDY state (ready to receive a command), the state of the BAI bit is immaterial, since no memory transfers will take place and incrementing the MXR SEL pointer will always occur during data buffer maintenance operation.**

### 4.15.11 RD ENA Pointer

The RD ENA pointer is primarily used for the read operation to allow the RC and RD registers to be clocked at the appropriate times.

During write-check or write operations, the pointer toggles but is not examined, and in data buffer maintenance operations, the pointer is alternately toggled as individual words are being sequenced through the buffer.

The RD ENA pointer is direct-set and direct-cleared in the same manner as the MXR SEL flip-flop just described. This pointer is also wired to toggle on alternate clock inputs. The clock input is MDPA REQ CLK H, which is the ORed condition of the RC and RD Clock signals (MDPA CLK RC H and MDPA CLK RD H). During read operations, there are two cases where the RD ENA pointer is inhibited from toggling. One is the setting of the Bus Address Increment Inhibit [CSTA BAI (1) H] bit, which causes data transfers to or from the same memory location and causes the RC or RD register (but not both), to be utilized. The second occurs when the DBL (Double) flip-flop is set. In this case, when doing double-word transfers to memory, the RD ENA pointer is overridden by AWRA EN DBL H. Conceptually, this can be thought of as adding two to the flip-flop pointer, which leaves it in its original state since two words are clocked simultaneously (one in RC and one in RD).

### 4.15.12 RB ENA Pointer

The RB ENA pointer is used during all operations (read, write, write-check and data buffer maintenance). It is direct-set and direct-cleared in the same manner as that previously described for the MXR SEL pointer. This flip-flop pointer is wired to toggle on alternate clock inputs and to control the clocking of the RA and RB registers. The clock input to this pointer is the MDPA INC RB ENA H signal, which is derived from the logic that generates the clock signals for the RA and RB registers. Consequently, when the MDPA CLK RA signal is asserted, the MDPA INC RB ENA H signal is asserted, causing the RB ENA pointer to switch and point to the odd-word RB register. When the CLK RB signal is generated, MDPA INC RB ENA H is again asserted and causes the RB ENA pointer to toggle to the even-word RA register.

If a write or write-check operation is being performed, the RB ENA pointer is inhibited from toggling if a double word from memory is transferred to RA and RB simultaneously. The pointer is overridden in this case and can conceptually be thought of as being incremented by two, which effectively leaves the flip-flop pointer in its original state. The top two-input AND gate in AND-OR gate E039-pin 6 accomplishes this function.

If the RH70 is not in the RDY (ready) state (performing a read, write, or write-check operation), and the BAI (Bus Address Increment inhibit) bit is set, the pointer is inhibited from toggling, since it is desired to transfer data to or from the same memory location. The lower two-input AND gate in AND-OR gate E039-pin 6 performs this function.

## 4.16 LOGIC DIAGRAM MDPE

This logic diagram contains the logic for clocking the RE register, the RE FULL flip-flop, data parity circuit and the exclusive-OR network used to compare the MXR contents with the OBUF contents during a write-check operation.

### 4.16.1 MDPE CLK MXR WORD H

The MDPE CLK MXR WORD H signal is used in data buffer maintenance operations, write and write-check functions. During data buffer maintenance and write operations, MXR data is clocked into the RE register as a result of MDPE CLK MXR WORD H. During write-check operations, the WCE (Write-Check Error) flip-flops are clocked by MDPE CLK MXR WORD H. In read operations, the RE register accepts data from the Massbus and the clocking logic is consequently inhibited for a Read command.

The MDPE CLK MXR WORD H is derived from delay line E011 and driver Q1. This input to the delay line is from OR-AND gate E028-pin 8.

**4.16.1.1 Write Operation** – For a write operation, this gate is qualified as follows:

1. The top two two-input gates are asserted when the MXR SEL flip-flop is pointing to the RD register [MDPD MXR SEL (1) L] and the RD FULL flip-flop is full [MDPA RD FULL (1) L] or when the MXR SEL flip-flop is pointing to the RC register [MDPD MXR SEL (0) L] and the RC FULL flip-flop is full [MDPA RC FULL (1) L]. These two gates, in essence, are asserted when the MXR SEL pointer is pointing to the register that is full.

2. During a write, CSTA WR L is asserted, qualifying the three-input gate.

3. RE FULL (0) L must be asserted, since it is necessary to have a word in the MXR and no word in the RE register to transfer a word to RE. This is accomplished in the four-input gate of E28.

With the above conditions satisfied, OR-AND gate E028-pin 8 is asserted and the pulse is driven down delay line E011. 100 ns after this, MDPE CLK MXR WORD H is asserted which asserts MDPE CLK RE L. This signal clocks the MXR data into the RE register. This pulse is 50-ns long and is turned off at the 150-ns tap on the delay line. Also at 150 ns, MDPE INC MXR SEL H is asserted and is ANDed with the state of the MDPD MXR SEL flip-flop to clear the RC or the RD register. For example, if data was transferred from the RD register to RE, MDPD MXR SEL (1) H is asserted, and when the MDPE INC MXR SEL H signal is asserted, MDPE CLR RD L is asserted, indicating that the contents of the RD register has been transferred to RE; the RD FULL flip-flop can now be cleared. This logic is accomplished for RD in AND gate E038-pin 8 and for RC in AND gate E038-pin 6. The MDPE CLR RC and MDPE CLR RD signals are 50-ns long as they are turned off at the 200-ns tap on delay line E011. On the trailing edge of the MDPE INC MXR SEL H signal, the MXR SEL pointer on MDPD is toggled to switch from the RC to RD or from the RD to RC register.

**4.16.1.2 Data Buffer Maintenance Operation** – During data buffer maintenance operation, the MDPE CLK MXR WORD H signal is generated in the same manner, except that in the third OR gate from the top in E028-pin 8, MDPE RDY L is asserted instead of CSTA WRITE L. The RDY signal is asserted when the RH70 is not doing a data transfer command.

**4.16.1.3 Write Check Operation** – During write-check operation, the delay line driven by the output of E028-pin 8 is asserted under the following conditions:

1. The two two-input gates of E228-pin 8 function the same as that described for write; in other words, they are asserted when the MXR SEL pointer is pointing to a register that is full.

2. The three-input OR gate is qualified when MDPD OBUF FULL (1) L is asserted. This is required since MDPE CLK MXR WORD H in a write-check operation is used to clock the WCE flip-flops and these flip-flops monitor the XOR comparision of the MXR and OBUF.

3. CSTA WR CHK L must be asserted for a write-check operation which qualifies the four-input gate to E028.

When the above conditions are satisfied, the MDPE CLK MXR WORD H is asserted similar to that described for the write operation.

### 4.16.2 MDPE CLK RE L Signal

The MDPE CLK RE L signal is generated by MDPE CLK MXR WORD H in a write operation and data buffer maintenance operation. Note that the output of AND gate E010-pin 6 is MDPE CLK MXR WORD H and is applied to AND-OR gate E039-pin 8, qualifying that gate to assert MDPE CLK RE L (except for a write-check operation).

During a read or write-check operation, data is taken from the drive to be loaded into RE. In these cases, the Drive Clock (CSTB DRIVE CLK H) signal qualifies AND-OR gate E039-pin 8 to assert MDPE CLK RE L. Drive Clock occurs on the trailing edge of SCLK and is a 50-ns pulse.

### 4.16.3 Inhibiting Delay Line E011

During a Read command, the delay line is inhibited since Massbus data rather than MXR data is clocked into RE. This is accomplished in AND gate E030-pin 8 by CSTA RD L.

During a write-check operation, if a write-check error occurs, the delay line is inhibited by the assertion of MDPE WCE L which inhibits AND gate E030-pin 8. This allows the data in OBUF to be frozen that so the programmer can examine it.

### 4.16.4 RE FULL Flip-Flop

The RE FULL flip-flop is clocked-set with assertion of the MDPE CLK RE L signal, indicating that the RE register is full. This flip-flop is cleared by the general-purpose CSTD DB INIT L or by the MDPD CLR RE L signal. After the data in RE is transferred to RF, MDPE CLR RE L is asserted, indicating that the data has been transferred to RF and the data in the RE register in now invalid.

### 4.16.5 MDPE REG FULL H Signal

The MDPE REG FULL H signal is used to flag the programmer of Data Late (DLT) conditions. During read or write-check operations, data from the Massbus is applied to the RE register. If the RE register is full [MDPE RE FULL (1) L] or if the RH70 is in the process of clearing the RE FULL flip-flop (MDPD CLR RE L), MDPE REG FULL H is asserted, meaning that data cannot be accepted by the RE register at that time.

**NOTE**

**The purpose of ORing CLR RE with RE FULL is to cover the condition where the RH70 is in the process of clearing RE FULL at the same time a clock pulse may be applied, which would normally cause the clock pulse to be lost.**

If the next Drive Clock signal (derived from the trailing edge of SCLK) occurs while MDPE REG FULL H is asserted, a DLT condition is flagged, indicating that the drive is ready to send another word to RE before the last word in RE has had time to be transferred to RF. This is shown on logic diagram CSTB.

### 4.16.6 Exclusive-OR Network

In write-check operations, 16 Exclusive-OR gates compare the contents of the MXR with the contents of OBUF. The MXR data may be either from the RC register if the data is an even word or from RD if the data is an odd word.

The outputs of the Exclusive-OR gates are open-collector ORed. If any two bits mismatch, the output line is pulled low. The four jumpers (W1, W2, W3, and W4) are provided as a maintenance aid to isolate a defective XOR gate on one of the integrated circuits (ICs). If a defective gate in an IC is pulling the output line low, the defective IC can be found by removing the associated jumpers, which isolate that IC from the output.

### 4.16.7 Odd Word and Even Word Write-Check Error

The output of the Exclusive-OR network is applied to the D inputs of the WCE OW (Write-Check Error–Odd Word) and WCE EW (Write-Check Error– Even Word) flip-flops. One of these flip-flops is clocked during a write-check command when MDPE CLK MXR WORD H is asserted. If the MXR SEL flip-flop is set, indicating an odd word in a double-word block, the WCE OW flip-flop is set and if the MXR SEL flip-flop is reset, indicating an even word in a double-word block, the WCE EW flip-flop is set when a write-check error occurs. To find out which memory word caused the write-check error, it is necessary to examine the Bus Address register which will reveal the last double-word block transferred. By implementing the logic with two WCE flip-flops (each having an associated status indicator in the CS3 register), the programmer can determine whether the write-check error was caused by the odd word or the even word in that double-word block.

The output of the WCE flip-flops are ORed in E019-pin 11 to assert MDPE WCE L, indicating the existence of a write-check error. The summary WCE bit can be read from bit 14 of the CS2 register.

### 4.16.8 Parity Checkers

Two nine-bit parity checker/generators are employed which examine the data in the RG register. During a write operation when the data in RG is transferred to OBUF, parity is computed and if the number of 1s in the data word is even, a parity bit is generated to OBUF so odd parity can be transferred to the Massbus. This parity bit is designated MDPE DATA PA L.

During read or write-check operations, a Massbus parity bit (MDPE RGPA L) is applied to the parity checker/generators along with the Massbus data. This Massbus parity bit has sequenced to the RG register from the RF and RE registers, at which point the signal name is MBSC SYNC PA.

**NOTE**

**The Massbus data is 18-bits wide and the RH70 data is 16-bits wide. As a result, bits 16 and 17 are X-ORed into one input of the parity circuit (MDPF RG1617 H).**

If the sum of the data plus parity (RG00–RG1617, plus RGPA) is odd, MDPE DATA PA L is asserted and no error is registered. If the sum is even DATA PA is negated (high) at the D input of the MDPE Massbus (Data Parity Error) flip-flop.

When data is clocked into OBUF (MDPD CLK OBUF H), except during a write, the MDPE flip-flop is clocked-set, registering a data parity error. Once the flip-flop is clocked-set, the 0 output inhibits the clock input, keeping the flip-flop locked until the error is recognized and cleared by the CSTB CLR ERR L signal.

The CLR ERR signal is a function of program clear, GO clear, Init, or DC LO (see logic diagrams CSTB and CSTD), or by posting a 1 in the TRE bit position of the CS1 register (error clear).

The Parity Test (PAT) bit is a control bit from the CS2 register which is used to invert the sense of the parity detection generation to allow the parity checker/generator circuits to be exercised.

### 4.16.9 RE Register Chip

The RE register chip shown on this diagram is a quad two-input mulitplexer and register which accepts Massbus data during read or write-check operations, or accepts MXR data during write or data buffer maintenance operations. Consequently, when MDPF RE SEL L is asserted, the A0 – A3 inputs to the multiplexer are enabled. These inputs accept Massbus data bits MBSC SYNC D17 H and MBSC SYNC D16 H and Massbus Parity bit MBSC SYNC PA H. The fourth input (A3) is tied high to provide for the logic inversion through the Exclusive-OR gate E018-pin 11 of the parity bit. The two Massbus data bits (D16 and D17) are Exclusively-ORed to yield MDPE RE1617 H, since these data bits from the Massbus represent the extra two bits not used (18 Massbus data bits versus 16 bits of data in the buffer). The two bits are monitored, however, to sense their parity. As a result, when the data is transferred to the RG register,an 18-bit parity check can be performed.

During write or data buffer maintenance operations, MDPF RE SEL L is unasserted which enables the B0 – B3 inputs to the multiplexer.

These inputs are tied to the MXR circuit. Note that B0 and B1 are grounded since the MXR is only 16 bits wide. The B2 and B3 inputs represent the byte parity for each word. These parity bits are Exclusively-ORed at the multiplexer output, resulting in word parity designated MDPE REPA L. The MDPE CLK RE L signal clocks the selected multiplxer inputs into the RE register which is part of this IC chip. Note that the output of Exclusive-OR gate E18-pin 11 (MDPE REPA L) is defined low, even though no logical inversion occurs between this signal and the input to the RE multiplexer/register. Conceptually, to maintain odd parity when converting byte parity to word parity via X-ORing of the byte parity bits, it is necessary to invert the sense of the resultant word parity bit.

## 4.17 LOGIC DIAGRAM MDPF

This diagram shows the RE register (except for the RE register chip shown on diagram MDPE), the RF register, the RG register and OBUF. These registers comprise the single-word section of the data path. The RE register multiplexer accepts data from the Massbus during read or write-check operations, or accepts MXR data during write or data buffer maintenance operation. The RE register consists of quad two-input multiplexer chips and a four-bit register. The A0 – A3 inputs are enabled during read or write-check operations by CSTA RD + WR CHK H. During write or data buffer maintenance operations, the B0 – B3 inputs are enabled. MDPE CLK RE L clocks the selected input data into the RE register.

The output of the RE register is fed to the RF register. In addition, the two Massbus data bits (which were Exclusively-ORed on diagram MDPE) and the parity bit are fed to the RF register. The RF register is clocked when the RF FULL flip-flop is set.

The output of the RF register is applied to the RG register when the MDPD CLK RG H signal is asserted. The output of the RG register is applied to the OBUF register and to the parity checkers on diagram MDPE when the MDPD CLK OBUF H signal is asserted. The OBUF register is the only register with a clear signal designated CSTB CLR OBUF DATA L. This signal is generated as a result of drive word count overflow or an error condition during a write operation; it allows the OBUF register to be cleared to zero-fill the rest of the sector (for disks) or record (for magtape). By storing the parity bit (OBUF PA) in the inverted sense, it is possible to receive correct (odd) parity when this clear signal is used.

The output of the OBUF register is applied to the Massbus drivers, (see logic print MBSA, MBSB, and MBSC) during a write operation, is applied to the IMX for the assembly of double words during a read operation, or is applied to one input of the XOR network in a write-check operation.

The output of the OBUF register is also supplied to the Unibus via the open-collector multiplexers shown on logic print CSTE for maintenance operation when reading the data buffer.

## 4.18 LOGIC DIAGRAM MDPH

This logic diagram contains the IMX and the MBCBUS data drivers for the RH70/Cache Interface. The IMX multiplexes OBUF data or Unibus data for transfer to the AMX or BMX which feeds the RA and RB registers.

During data buffer maintenance operations, the Unibus data is supplied to IMX; during read operations OBUF data is supplied to the IMX.

The right half of this diagram shows the 32 data drivers and the four parity drivers for the RH70/Cache Interface. Note that the contents of the RC register are supplied to data lines D00 – D15 (even word) and the contents of the RD register are supplied to data lines D16 – D31 (odd word). These lines accept data from the RC and RD registers for transfer to memory during a read operation. The data and parity drivers are consequently enabled by MDPH READ H and the appropriate SEL DATA signal from the Cache. This diagram is common to all four RH70 Controllers (Controller A, B, C, or D) and the signal is designated CDPJ SELDATA CNTLX H. When the RH70 makes a request to Cache, the Cache arbitrates this request with all other requests. If this module is plugged into the Controller D slot for example, the Cache will send a CDPJ SELDATA CNTLD H signal to this module, which will allow the data in the data drivers from this module to be transferred to memory. Each of the four RH70 slots in the CPU backplane has a separate wire from the Cache, which allows Cache to select one of the four controllers by one of the four SEL DATA CNTL signals.

## 4.19 LOGIC DIAGRAM CSTA

This diagram contains the command code decoding logic, various control and status bits, the FCTN LOAD, BAI, and BUSY flip-flops.

### 4.19.1 Command Code Decoding Logic

The command code is represented in the least significant bits of the CS1 register as shown below.

| D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|
| F4 | F3 | F2 | F1 | F0 | G0 |
| | MSD octal digit | | | LSD octal digit | |

5=write-check
6=write
7=read

The three data transfer command codes (read, write, and write-check) are decoded by BCD-to-decimal decoder E56. This decoder decodes data bits D03 – D05 from the Unibus data lines. If a Write-Check command is specified, a binary code of 5 is supplied to the decoder; the F5 output is asserted and fed to the data input of the WR CHECK flip-flop. If a Write command is specified, a binary code of 6 is supplied to the decoder, the F6 output is asserted and fed to the data input of the WRITE flip-flop. If a Read command is specified, a binary code of 7 is supplied to the decoder, the F7 output is asserted and fed to the data input of the READ flip-flop. The GO bit must be asserted for the hardware to execute the command requested by the program. The command flip-flops are clocked by the trailing edge of the GO pulse if the RH70 is in the ready (RDY) state. At the completion of the command execution, the flip-flop which had been set will be cleared by the assertion of BUSY (0) H, if a register is not currently being accessed by the program (DEV SEL unasserted). This is accomplished by gate E46 pin 6.

Various gates are shown on the outputs of these flip-flops and provide buffered versions of the signals and also provide combinations of these signals for use by the data buffer control logic.

### 4.19.2 GO Bit, GO CLR Pulse

The GO Bit is stored in the drive in bit 00 of the CS1 register and must be asserted for the execution of all data transfer commands. This results in all data transfer commands having odd command codes. For example, the data transfer command codes for a write-check operation are 51, 53, 55 and 57. In addition, all reverse operations are specified by a 7 in the least significant digit. For example, the read reverse command code is 77. The CSTA GO H signal is asserted when the following conditions are satisfied:

1. A data transfer command is specified which enables one of the inputs to gate E52-pin 6, depending on the command specified.

2. BCTD D00 IN H is asserted which designates the GO bit.

3. The signals associated with writing the low byte of the CS1 register are asserted. These signals are BCTB REG STR H, BCTA CS1 IN L, and BCTA LO BYTE H.

If the three data transfer command flip-flops are all reset, indicating that the RH70 is in the ready state (RDY asserted) the GO bit asserts CSTA GO CLR L via NAND gate E62-pin 8. GO CLR initializes the data path, clears all error conditions, and sets up for a data transfer operation. In addition, the GO CLR pulse sets the BUSY flip-flop, which indicates that a data transfer command is being executed.

### 4.19.3 Function Load Flip-Flop

The Function Load (FCTN LOAD) flip-flop is set when a data transfer command code is being loaded. The trailing edge of the GO CLR signal clocks the FCTN LOAD flip-flop set. The BCTA DEV SEL H signal is asserted, indicating that the RH70 is selected to do a register read or write and holds the direct clear input high, preventing FCTN LOAD from resetting. The RH70 can monitor for non-existent devices this time, or can request the first word from memory during write or write-check operations. When the RH70 is deselected (DEV SEL unasserted), the FCTN LOAD flip-flop is direct cleared.

### 4.19.4 Ready State (RDY)

When the READ, WRITE and WR CHECK flip-flops are cleared, the CSTA RDY H signal is asserted, indicating that the RH70 is not doing a data transfer command.

The 0 outputs of these flip-flops are ANDed in AND gate E57-pin 6 and AND gate E50-pin 12 to assert RDY.

### 4.19.5 BUSY Flip-Flop

The BUSY flip-flop is set when the RH70 is doing a data transfer command. When a data transfer command is specified, it is transferred to the drive with the GO bit set. The GO bit creates a GO CLR, which directs-sets the BUSY flip-flop. Normal completion of a data transfer operation occurs upon receipt of a clock pulse with the data input grounded to clear BUSY. When the drive reaches the End of Segment (EOS) with the RUN line unasserted, CSTB EOS (1) H is set. This indicates that the RH70/Massbus transfer is complete. When CSTC DONE H occurs (indicating completion of the RH70/Memory data transfer), along with EOS (1) H, the BUSY flip-flop is clocked clear, indicating termination of the data transfer. The BUSY (0) H signal is supplied to the READ, WRITE and WR CHECK flip-flops and clears them.

The BUSY flip-flop is also direct-cleared under the following conditions:

1. CSTD CLR L, which is a function of initializing the RH70, power up, etc.

2. CSTB MXF L, which is an error condition in the CS2 register, indicating that the hardware failed or that the data transfer did not complete in the normal manner. This error condition forces termination by direct-clearing the BUSY flip-flop.

3. CSTA FCTN LOAD (1) H and CSTB NED H. These two signals are asserted when a data transfer command is attempted to be loaded into a non-existent device. Since there will be no response from the device, the data transfer is terminated by direct-clearing the BUSY flip-flop.

### 4.19.6 Invert Parity Check (IPCK)

CSTA IPCK0 - CSTA IPCK3 are the four least significant bits of the CS3 register and are used to invert parity checking on the data words received from memory during write and write-check operations. IPCK0 is the bit associated with the even word–low byte, IPCK1 with even word–high byte, IPCK2 with odd word–low byte, and IPCK3 with odd word–high byte.

The IPCK0 - IPCK3 bits are applied to the open-collector BUSI multiplexers on logic print CSTE (which may be read when reading the CS3 register). The IPCK0 - IPCK3 bits are also ANDed with CSTA READ (0) H to create invert signals CSTA INV0 H - CSTA INV3 H, respectively. When a read operation (transferring data to memory) is specified, parity should not be inverted. Consequently, the CSTA READ (0) H is unasserted (driven low) during a read operation, inhibiting the invert signals. For any other operation, the invert signals are enabled. During data buffer maintenance operation, the IPCK bits invert the parity bits generated when moving data through the RSDB.

### 4.19.7 Unit Select Bits and Program Clear (PG CLR)

The Unit Select bits are three bits which specify one of eight drives. They are located in bits 0 - 3 of the CS2 register and drive the Drive Select (DS00–DS03) lines on the Massbus. The unit to be selected is specified by BCTD D00 IN H - BCTD D03 IN H from the Unibus. The Unit Select bits are clocked by BCTA CS2 IN L, BCTA LO BYTE H, and BCTB REG STR. These signals are ANDed in AND gate E54-pin 8. The output of the gate, in addition to clocking the Unit Select bits, also asserts CSTA PG CLR L if BCTD D05 IN H is asserted. D05 IN H is asserted when the program writes a 1 in bit 5 (CLR) of the CS2 register. The PG CLR signal clears the RH70 and the associated drives.

### 4.19.8 PAT Bit

The Parity Test (CSTA PAT H) bit controls the parity generator between the RG and OBUF registers in the data buffer (see logic diagram MDPE) and inverts parity associated with Massbus data. Inverting the PAT bit during a write operation will convert odd (correct) parity to even (wrong) parity

to allow exercising of the drive's parity checking circuits. Inverting the PAT bit during a read or write-check operation will convert odd parity to even parity to allow the exercising of the RH70 parity checking network. The PAT bit is clocked by the same signals which clock the Unit Select bits.

The PAT bit also controls the generation of the parity bit associated with the control lines on the Massbus. When PAT is set, even (wrong) parity is sent along with the data over the control data lines when writing a remote register. During data buffer maintenance operation, the PAT bit will invert the parity checking for data words bubbling from the RG register to OBUF.

### 4.19.9 Bus Address Increment Inhibit Flip-Flop

The Bus Address Increment Inhibit (BAI) flip-flop, when set, inhibits the incrementing of the bus address. The BAI flip-flop is clocked by BCTA CS2 IN L, BCTA LO BYTE H, and BCTB REG STR H, which are ANDed in AND gate E54-pin 8. The RH70 must be in the RDY state (not performing a data transfer command) in order for the BAI flip-flop to be clocked. This ensures that the BAI flip-flop will not change state during the middle of a memory transfer, which could cause a spike on the address lines. BAI is loaded from data bit D03 on the Unibus and is cleared by the general-purpose CSTD CLR L signal (INIT, DC LO, or Program Clear-bit 5 of CS2).

## 4.20 LOGIC DIAGRAM CSTB

This diagram contains the Massbus control logic, the interrupt logic, and the various error conditions.

### 4.20.1 Massbus Control Logic

To effectively understand the Massbus control logic, the following signals are reviewed. When transferring data to the drive (write operation), the RUN line is asserted on the Massbus, provided that the required number of words are in the data buffer (four memory cycles). When transferring data from the drive (read or write check operations), the RUN line on the Massbus is asserted immediately, indicating that the RH70 is prepared to receive data from the drive.

At the end of each logical block of data transferred to or from the drive, the drive asserts an EBL (End-of-Block) pulse. At the trailing edge of EBL the drive monitors the RUN line. If it is asserted, it will continue a data transfer of the next logical block. If RUN is unasserted, the data transfer operation is terminated. The transfer of data to or from the drive is accomplished by drive-generated SCLK signals. In a read or write-check operation, the drive provides a data word on the Massbus, on the leading edge of SCLK, and the RH70 strobes the data in its buffer on the trailing edge of SCLK. In a write operation, the RH70 places a data word on the Massbus, on the trailing edge of SCLK and the drive strobes it into its buffer on the leading edge of SCLK. The SCLK signal from the drive to the RH70 is rerouted back to the drive as a WRITE CLK signal. WRITE CLK is used during a write operation to cause the drive to clock the data from the Massbus into its buffer.

At the trailing edge of SCLK, a 50-ns pulse, called DRIVE CLK, is generated unless drive word count overflow or an error condition occurs. This pulse indicates that the RH70 is going to receive or change data.

**4.20.1.1 RUN Flip-Flop** – The RUN line is asserted on the Massbus as a result of setting the RUN flip-flop. This flip-flop is set under any of the following conditions:

1. When a write operation is specified, a word is in OBUF and the required number of words are stored in the data buffer. The CSTA WRITE (1) H signal denotes that a write command has been loaded, the MDPD OBUF FULL (1) H flip-flop sets when a data word is in OBUF, and CSTE START H is asserted when four memory cycles have taken place. Four memory cycles could mean as few as four words in the data buffer (four single-word transfers) or as many as eight words (four double-word transfers). Typically, the data buffer is designed for double-word operation and will contain seven or eight data words. For example, if the first word to be transferred is on a single-word boundary, one word would be transferred on the first memory cycle and double words are then transferred, resulting in a data buffer containing seven words. CSTE START H, CSTA WRITE (1) H, and MDPD OBUF FULL (1) H are ANDed in gate E40-pin 6 and clock the RUN flip-flop set, which asserts CSTB RUN H at the output of Inverter E43-pin 10.

2. When a read [CSTA READ (1) L] or write-check [CSTA WR CHECK (1) L] operation is specified, the RUN flip-flop is clocked set immediately, indicating that the RH70 is prepared to accept data.

3. If an error is detected in the RH70 before the START signal is asserted, the Transfer Error (TRE) condition clocks the RUN flip-flop set and forces the RUN line asserted, which allows the drive to normally end the command execution.

The RUN flip-flop is cleared under the following conditions:

1. The TRE condition is applied to the set input of the DIS SCLK flip-flop. The clock input to this flip-flop is the SYNC CLK signal. When SYNC CLK is asserted, DIS SCLK is set, which disables SYNC CLK signals and causes the RUN flip-flop to be direct-cleared, which in turn negates CSTB RUN H.

2. When the drive detects an error, it asserts the EXC (Exception) line. This signal is ANDed with EBL and causes the CSTB EXC SAVE flip-flop to set. CSTB EXC SAVE (1) direct-clears the RUN flip-flop via gate E25-pin 12.

3. When the required number of words have been transferred on the Massbus, the drive word count overflow [AWRE DRWC OFLO (1) L] flip-flop is set, which direct-clears the RUN flip-flop.

4. When the RH70 is not busy [CSTA BUSY (0) L], the RUN flip-flop is direct-cleared. The conditions under which the BUSY flip-flop are cleared are:

   a. Normal data transfer termination. CSTB EOS (1) H and CSTC DONE H are asserted.

   b. General-purpose clear signal (CSTD CLR H).

   c. Missed transfer error occurs.

   d. When the RH70 attempts to load a data transfer command into a nonexistent drive.

**4.20.1.2 End of Segment (EOS) Flip-Flop** – The drive monitors the RUN line at the trailing edge of EBL and if the RUN line is unasserted, the drive will stop transferring data. The RH70 also detects this condition and when detected, the CSTB EOS flip-flop is set. Note that the RUN signal is applied to the D input of EOS. The EBL pulse is applied to the clock input of the flip-flop. The positive-going trailing edge of EBL sets the flip-flop if RUN is unasserted. The setting of EOS indicates termination of the Massbus data transfer. CSTB EOS (1) H is ANDed with CSTC DONE H, indicating termination of the RH70/Memory data transfers. These conditions clear BUSY, which denotes that the data transfer is completed on the Massbus and on the Memory bus. The EOS flip-flop is cleared by the general-purpose CLR signal, or by GO CLR, which is asserted when a data transfer command has been loaded in the drive.

**4.20.1.3 Disable Sync Clock (DIS SCLK) Flip-Flop** – When the DIS SCLK flip-flop is set, it indicates that the data transfer has been completed or that an error condition has occurred, in which case, it is desired to inhibit SYNC CLKs and to terminate the transfer. A Data Late (DLT) condition causes DIS SCLK to be direct-set since the DLT condition is detected at the time of SYNC CLK. It is therefore necessary to immediately set DIS SCLK. DIS SCLK can also be set by a transfer error [CSTB TRE (1) L] or by drive word count overflow [AWRE DRWC OFLO (1) L]. These signals are ORed in E45-pin 8 and applied to the D input of DIS SCLK. The CSTB SYNC CLK signal is applied to the clock input and the leading edge of SYNC CLK sets DIS SCLK if either condition is present. DIS SCLK is cleared by the general-purpose CLR signal, or by the GO CLR signal, which is asserted when a data transfer command has been loaded into the drive.

**4.20.1.4 DRIVE CLK Signal** – The DRIVE CLK signal is a 50-ns pulse used by the RH70 and is initiated on the trailing edge of SYNC CLK. The circuitry that generates it is Driver E22-pin 8, 50-ns delay line DL2, and Receiver E17-pin 13. To understand the operation, assume that DIS SCLK (0) H is asserted and the MBSA SYNC CLK H signal is present. These conditions turn Driver E22 on. The low output of the driver is applied to the delay line.

The DRIVE CLK signal, however, is not generated since the SYNC CLK H signal inhibits pin 11 of Receiver E17. After a period of time, the SYNC CLK signal is driven low by the drive. Now, the driver is turned off and the receiver is enabled, producing CSTB DRIVE CLK H.

When the driver is turned off, a positive-going transition appears at its output. This transition is fed to the receiver after 50 ns (the time required to propagate down the delay line). At this point, the receiver becomes inhibited again and the DRIVE CLK signal becomes unasserted. This results in a 50-ns pulse initiated at the trailing edge of SYNC CLK. If DIS SCLK (0) H is unasserted, the driver is inhibited, which in turn inhibits the receiver, thereby preventing generation of the DRIVE CLK signal.

**4.20.1.5 Zero Filling the Sector or Record** – During a write operation, it is necessary to gate the data from OBUF onto the Massbus data lines. This is accomplished by the CSTB GATE SYNCD H signal. If an error condition [CSTB DIS SCLK (1) L] occurs, or if drive word count overflow [AWRE DRWC OFLO (1) L] occurs before the end of a segment (disks) or a record (magtapes), the rest of the segment or record is filled with zeros. This is accomplished by clearing the OBUF register at the trailing edge of SYNC CLK when either of these conditions exist. The logic that accomplishes this is gates E39-pin 11 and E40-pin 8.

### 4.20.2 Non-Existent Drive (NED) Flip-Flop

When the program reads or writes a remote register, the RH70 asserts Demand on the Massbus. Within 1.5 $\mu$s, the drive should respond with TRA (transfer). If not, a 1.5 $\mu$s one-shot multivibrator times out and asserts BCTB SET NED L. This signal is fed to a latch which generates CSTB NED H, indicating that a non-existent drive has been accessed. CSTB NED H is applied to the clock input of the TRE flip-flop via gate E20-pin 8 and clocks TRE set, indicating that a transfer error has occurred.

NED is cleared by the CSTB CLR ERR B L signal which is a result of Unibus INIT, Power Fail, setting bit 5 (CLR) in the CS2 register, or posting a 1 in bit 14 (TRE) of the CS1 register (error clear). NED is bit 12 in the CS2 register.

### 4.20.3 Program Error (PGE) Flip-Flop

The PGE flip-flop is set when the program attempts to load a data transfer command while a data transfer command is already in progress. CSTA RDY is applied to the D input of the flip-flop and is unasserted when the RH70 is not ready (in the busy state). The CSTA GO H signal is applied to the clock input, which indicates that an attempt has been made to load a data transfer command with the GO bit set. When this occurs with the RH70 in the busy state, PGE sets. This causes CSTB STOP DEM L to be asserted. This signal prevents the data transfer command from being loaded on the Massbus.

The PGE flip-flop is cleared by the CLR ERR signal as a result of Unibus INIT, Power Fail, loading a 1 in bit 5 (CLR) of the CS2 register, or posting a 1 in bit 14 (TRE) of the CS1 register. PGE is bit 10 in the CS2 register.

### 4.20.4 Missed Transfer (MXF) Latch

When a command is loaded into the drive and the drive recognizes this command as one that it implements, it sends an OCC (Occupied) signal to the RH70. This signal remains asserted until the last EBL pulse, or until an EBL pulse with the RUN line unasserted is received. When the RH70 becomes busy, CSTB RDY L becomes unasserted, providing a high level on pin 10 of 650 $\mu$s one-shot multivibrator E27. With no SYNC CLK signal applied, pin 9 of the one-shot is low. These conditions fire the one-shot and it begins to time-out for 650 $\mu$s. However, when the RH70 loaded a command into the drive and the drive recognized the command, it returned OCC. This signal is applied to the one-shot to terminate it and is also applied to the CSTB set MXF flip-flop, keeping it direct-cleared. In this case, no MXF (missed transfer) condition occurs. If the drive did not respond with the OCC signal, the one-shot would time-out. After 650$\mu$s, the positive-going transition from the 0 output of the one-shot is applied to the SET MXF flip-flop. Since this flip-flop is not held direct-cleared by OCC, it sets and causes CSTB MXF H to be asserted. This signal is applied to the TRE flip-flop via gate E20-pin 8, causing TRE (Transfer Error) to be raised (bit 14 of the CS1 register). This operation describes the time-out feature associated with the initiation of a data transfer command.

A similar time-out feature is present during the completion of a data transfer command and is implemented as follows. Assume that the RH70 has loaded a command, the drive has recognized the command and has returned the OCC signal, which remains asserted throughout the transfer. When the drive completes the transfer and reaches the last EBL (or EBL with the RUN line unasserted), it removes the OCC signal from the Massbus, thus removing the direct-clear from the one-shot and the SET MXF flip-flop. Since the inputs to the 650 $\mu$s one-shot are still enabled (low on pin 9, high on pin 10), the one-shot begins to time-out again. If the BUSY flip-flop is not cleared within 650 $\mu$s of the removal of OCC, the one-shot times-out and the positive-going transition at the output sets the SET MXF flip-flop. If the BUSY flip-flop is cleared within 650 $\mu$s of the removal of OCC, the one-shot is cleared and the SET MXF flip-flop is prevented from setting. MXF is bit 9 in the CS2 register.

### 4.20.5 Data Late (DLT SYNC) Flip-Flop

The Data Late synchronizing flip-flop provides for synchronization of data late conditions – i.e., to prevent clocking of data late conditions just as they occur. The DLT SYNC flip-flop causes the data late condition to settle for the width of Drive Clock. At the end of Drive Clock, the DLT SYNC flip-flop is sampled to see if a DLT condition has occurred. If it has, the DLT flip-flop is set.

A Data Late condition can occur as a result of the following:

1. During a read or write-check operation (when reading data from the drive), the drive attempts to put a word in the RH70 while a word is already stored where the data is to be transferred. In the case of Read or Write-Check commands (CSTA RD + WR CLK H), the words from the drive are transferred to RE and a Data Late condition occurs when the drive attempts to put a word in RE when RE is already full (MDPE REG FULL H). With these two conditions present, the DLT flip-flop is set on the trailing edge of DRIVE CLK.
2. During a write operation, where data is transferred from the OBUF register in the RH70 to the drive, a word must be available for transfer to OBUF from the RG register when the present word in OBUF is transferred to the drive. This information is implemented by MDPD RG RDY (0) H. Consequently, during a write operation [CSTA WRITE (1) H], if RG RDY (0) H is asserted (indicating a data word is not available for transfer to OBUF), the DLT flip-flop is set on the trailing edge of Drive Clock. In the case where the word in OBUF is the last word to be transferred, RD RDY (0) H will be asserted since there are no more words to transfer. Consequently, AWRE DRWC OFLO (0) H becomes unasserted, indicating that all words have been transferred onto the Massbus. This signal inhibits gate E50-pin 6 from setting the DLT SYNC flip-flop.

The DLT SYNC and DLT flip-flops are cleared by the general-purpose CSTB CLR ERR L signal. DLT is bit 15 of the CS2 register.

### 4.20.6 Exception Save (EXC SAVE) Flip-Flop

When the drive is doing a data transfer and has detected an error, it raises the Exception (EXC) line on the Massbus. The RH70 sets the EXC SAVE flip-flop at EBL time. MBSB EBL H and MBSB EXCP H are ANDed in gate E41-pin 3 and clock the EXC SAVE flip-flop set.

During a write or write-check operation, CSTA WRITE (1) L or CSTA WR CHK (1) L, respectively, is asserted. Either signal is ANDed with EXC SAVE in gate E46-pin 11 to yield the Exception Error condition (CSTB EXC ERR L). The error condition is asserted immediately in the write or write-check operation.

In a read operation, it is necessary to wait until the data buffer is emptied before the EXC ERR condition is asserted. This ensures that all good data in the data buffer, including the word which caused the posting of the error, has been transferred to memory. The signals that implement this are

MDPD DB EMPTY L and CSTB EXC SAVE, which are ANDed in gate E46-pin 11. When the CSTB EXCP ERR L signal is asserted, the transfer error (TRE-bit 14 in CS1) flip-flop is set.

### 4.20.7 Read Exception Circuitry

If a drive detects an error during a data transfer read operation, CSTA READ (1) H and CSTB EXC SAVE are asserted and yield CSTB READ EXCP H via AND gate E35-pin 8. READ EXCP controls the double-word logic shown on logic diagram AWRA. This is to force a single-word memory operation if only one word is in the data buffer (MDPD LAST WORD), the drive will not send more words (due to the Exception Error), and the RH70 is waiting to assemble a double word.

### 4.20.8 Transfer Error (TRE) Flip-Flop

The TRE flip-flop summarizes all RH70 and all drive error conditions. The RH70 error conditions are:

Write-check error - MDPE WCE H
Parity error - CSTC PE H
Data Late - CSTB DLT (1) L
Non-existent drive - CSTB NED L
Non-existent memory - CSTC NEM (1) L
Programming Error - CSTB PGE (1) L
Missed transfer - CSTB MXF (1) L
Massbus data bus
Parity error - MDPE (1) L

The drive error condition is reported as an exception error (CSTB EXCP ERR L). The TRE error conditions are summarized in the Special Conditions (SC-bit 15 of CS1) bit. The TRE flip-flop is cleared by the CSTB CLR ERR L signal which is asserted as a result of CSTD CLR . GO CLR L (see logic diagram CSTD) or by loading a 1 in the TRE bit position of the CS1 register. The gating for clearing TRE in this instance is CSTA CS1 IN H, BCTA HI BYTE H, BCTB REG STR H, and BCTD D14 IN H, which are ANDed in gates E59-pin 8 and E52-pin 12.

### 4.20.9 Special Conditions (CSTB SC H)

All conditions requiring action by the CPU are summarized in the Special Conditions (SC bit 15 in CS1) bit. This is accomplished by gate E44-pin 8 which ORs TRE (Transfer Error), ATTN (Attention condition from a drive on the Massbus), or MCPE (Massbus Control Bus Parity Error). The SC bit will cause an interrupt if the Interrupt Enable (IE) bit is set.

### 4.20.10 Interrupt Logic

The Interrupt Request logic, when enabled, causes CSTB INTR REQ L to be asserted. This signal is applied to the logic on logic diagram BCTC and initiates an interrupt to the CPU over the Unibus. An interrupt may be generated in one of the following ways:

1. A programmed interrupt which occurs by writing 1s into the IE and RDY bit positions of the CS1 register (bits 6 and 7, respectively). This is implemented by gate E52-pin 8 which direct-sets the INTR flip-flop. The CSTA CLK CS1 LO signal enables the low byte of the CS1 register.

2. Successful completion of a data transfer command. When the RH70 goes from busy to the ready state, CSTA RDY H clocks the INTR flip-flop.

3. A Special Condition (SC) which occurs as a result of the assertion of SC when IE (Interrupt Enable) is set with the RDY bit asserted (RH70 not busy). If the RH70 is busy, it is inhibited from causing an interrupt until the control is back into the Ready state. This logic is implemented in NAND gate E44-pin 6. In order to set the INTR flip-flop, the IE flip-flop must set. This flip-flop is set as a result of loading bit 6 from the low byte of the CS1 or CS3 registers during register strobe. The gating for this is shown at the input to the IE flip-flop.

   The IE and the INTR flip-flops are cleared by BCTC INTR DONE H or by CSTD CLR H. The INTR DONE signal indicates that the CPU has recognized the interrupt and has accepted the interrupt vector. The IE and INTR flip-flops are cleared by INTR DONE to prevent further interrupts during the interrupt service routine.

## 4.21 LOGIC DIAGRAM CSTC

This logic diagram contains the memory cycle control logic and memory cycle error logic, including data parity error, address parity error, and non-existent memory.

### 4.21.1 Memory Cycle Control Logic

The memory cycle is divided into three states: a Start Memory Cycle (SMC), a Request (REQ), and a Cycle-in-Progress (CIP). Initially, the memory cycle is started by MDPA START MEM H which occurs when the data in RA/RB is transferred to RC/RD (Figure 4-9). This signal clocks the SMC flip-flop, which sets to indicate that the RH70 is ready to make a memory request. The SMC flip-flop must be set in order for the RH70 to send a memory request to Cache.

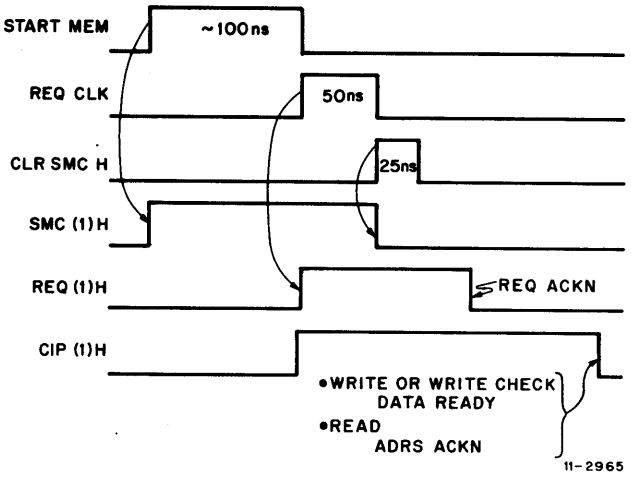



Figure 4-9 Memory Cycle Simplified Timing Diagram

Approximately 100 ns after SMC is clocked, MDPA REQ CLK H is generated via the delay line on the data buffer module. This signal clocks the REQ flip-flop set, indicating that a memory request is issued to Cache. The memory request is designated CSTC CNTLX REQ L. (As previously described, the X in the signal name designates one of four controllers.) The reset output of the REQ flip-flop is applied to the direct-set input of the CIP flip-flop and sets it. The setting of this flip-flop indicates that a memory cycle has been initiated. The CIP flip-flop remains set until the transfer is completed. Approximately 50 ns after REQ is set, the SMC flip-flop is reset by the MDPA CLR SMC H signal.

For a read operation (words written into memory), the memory cycle is completed when the Cache returns Address Acknowledge (CCBE MBC ADRS ACKN L). For write or write-check operations, the memory cycle is completed upon receipt of the Data Ready (CDPK DATA RDY CNTLX H) signal from Cache.

**4.21.1.1 SMC Flip-Flop** - The SMC flip-flop is clocked at MDPA START MEM H time. This signal is a timing pulse from the delay lines which clocks the data from RA/RB into RC/RD. The SMC flip-flop is set if there is no error [CSTB TRE (1) L], no word count overflow [AWRE WC OFLO (1) L], or no program clear or initialize operation [CSTD PG CLR + INIT (1) L] in progress. The three signals are ORed into the D input of the SMC flip-flop and are also fed through a 50-ns delay line.

This allows the SMC flip-flop to synchronize these conditions. If one of these conditions has been asserted for 50 ns and the memory cycle has not been started [SMC (0)], CSTC DONE H will be asserted (if the memory cycle has been started) one more memory cycle is accomplished before CSTC DONE can be asserted (see gate E17-pin 3 and gate E35-pin 6). When DONE is asserted, it indicates that the data transfer command has completed on the RH70/Cache Interface.

The SMC flip-flop is direct-cleared by the general-purpose CSTD CLR . GO CLR signal, or by MDPA CLR SMC H. CLR SMC occurs at the end of REQ CLK and is a 25-ns pulse. REQ CLK is a 50-ns pulse occurring at the end of SMC, which is a 100-ns pulse.

**4.21.1.2 REQ Flip-Flop** - The REQ flip-flop sets when the RH70 is ready to perform a memory request and indicates that the RH70 requests use of Cache. SMC must be set in order for REQ to set. With SMC in the reset state, the D input to REQ is held low. Once REQ is set, CSTC CNTLX REQ L is issued to Cache. The RH70 now waits for Cache to acknowledge the request. When Cache acknowledges the request with CCBE REQ ACKN, this signal is ANDed with CDPJ SELADRS CNTLX H to assert AWRD REQ ACKN H (see logic diagram AWRD). AWRD REQ ACKN H then direct-clears the REQ flip-flop.

NOTE

**If the 650 microsecond transfer error one-shot times out due to a hardware failure (see logic diagram CSTB), the time-out will clear the BUSY flip-flop, putting the RH70 back into the RDY state. In this case, CSTA RDY B H direct-clears REQ to ensure that the REQ flip-flop is not locked in the set state.**

There are several conditions which hold the D input to the REQ flip-flop low, thus preventing a memory request to Cache:

1. CSTC SMC (0) H – if this flip-flop is cleared, it drives the D input to REQ low, preventing the REQ flip-flop from setting.

2. If there is a data parity error associated with the odd word or the even word, the D input to REQ is driven low (see pins 9 and 3 of AND-OR gate E38-pin 8).

3. When the RH70 is in the data buffer maintenance mode of operation, memory requests are inhibited by the assertion of CSTB RDY L.

4. If a write-check error occurs, further memory requests are inhibited as a result of MDPE WCE H being asserted.

In a write or write-check operation, the timing to perform the memory cycle is a REQ CLK signal from the data buffer when data is clocked from RA/RB into RC/RD. However, for the first word transferred there is no REQ CLK signal. Consequently, when a data transfer command is loaded and a non-existent device has not been addressed, 45-ns one-shot E27 is triggered which asserts CSTC FST REQ (first request). This signal is applied to the direct-set input of the REQ flip-flop to force the flip-flop set for the first memory cycle.

**4.21.1.3 CIP Flip-Flop** – When the REQ flip-flop sets, the low-level 0 output direct-sets the CIP flip-flop, indicating that a memory cycle has started. The CIP flip-flop remains set for the rest of the memory cycle. The CIP flip-flop is used to indicate a DONE (CSTC DONE H) condition after the last memory cycle has completed.

For write or write-check operations, the CIP flip-flop is clocked clear on the trailing edge of CDPK DATA RDY CNTLX H. This is one of the gating signals from Cache used to generate CLK RA AND CLK RB signals for clocking data into the data buffer. For read operations, the CIP flip-flop is clocked by the trailing edge of CSTC READ DONE L. READ DONE is asserted upon receipt of CCBE MBC ADRS ACKN L from Cache during a read operation if this controller is selected. This logic is implemented by ANDing CCBE MBC ADRS ACKN L, CSTA READ (1) H, and CDPJ SELDATA CNTLX H in NAND gate E24-pin 6. Delay line DL1 converts ADRS ACKN into a 50-ns pulse.

NOTE

**The leading edge of the READ DONE signal clears the RC FULL and RD FULL flip-flops (see logic diagram MDPA) to allow new data to be sequenced into RC and RD.**

The CIP flip-flop is direct-cleared due to:

1. Address Parity Error [CSTC APE (1) L] on the memory address and control lines. In this error, the RH70 does not receive any timing signals from Cache so the CIP flip-flop must be direct-cleared.

2. Addressing Non-Existent Memory [CSTC NEM (1) L]. When this condition occurs, there are no timing signals from Cache so the CIP flip-flop must be direct-cleared.

3. RH70 going to Ready State (CSTB RDY L). If the 650 $\mu$s Missed Transfer error one-shot times-out due to a hardware failure, the BUSY flip-flop is cleared, forcing the RH70 back into the Ready state. This ensures that the REQ and CIP flip-flops are cleared so that they are not latched in the set state.

**4.21.2 Data Parity Error**

The data parity error logic contains two flip-flops and associated gating. One flip-flop is designated DPE OW (Data Parity Error – Odd Word) and is used during write or write-check operations to detect a parity error on the odd data word. The second flip-flop is designated DPE EW (Data Parity Error – Even Word) and is used during write or write-check operations to detect a parity error on the even data word.

If a single-word operation is specified, one of two enable signals is asserted. When an odd data word is being transferred during a write or write-check operation, MDPC EN OWPCK (Enable Odd Word Parity Check) is asserted. If one of the odd word parity bits, high byte or low byte (MDPC OWHB PAB H or MDPC OWLB PAB H) is raised (denoting an odd word parity error), the data input to the DPE OW flip-flop is asserted. At REQ CLK time, the flip-flop is clocked set, indicating an odd word high byte or odd word low byte parity error.

When an even data word is being transferred during a write or write-check operation, MDPC EN EWPCK (Enable Even Word Parity Check) is asserted. If one of the even word parity bits (MDPC EWHB PAB H or MDPC EWLB PAB H) is raised (denoting an even word parity error), the data input to the DPE EW flip-flop is asserted. At REQ CLK TIME, the flip-flop is clocked set, indicating an even word low byte or high byte parity error.

If double-word operation is specified, all four parity bits are monitored and both enable signals (MDPA EN OWPCK H and MDPA EN EWPCK H) are asserted. If any of the four parity bits is asserted high, it indicates a parity error and will cause either or both of the data parity error flip-flops to set, depending on the parity bit(s) asserted. When a parity error is detected, the output of AND-OR gate E38-pin 8 is driven low, which prevents the data input of the REQ flip-flop from being enabled, thus preventing further memory cycles from occurring. The DPE OW and DPE EW error conditions are indicated in bits 14 and 13 of the CS3 register and are ORed together in gate E24-pin 12 to cause a parity error (PE bit 13 of CS2 register). The data parity error flip-flops are cleared by the CSTB CLR ERR B L signal.

#### 4.21.3 Address Parity Error (APE), Non-Existent Memory (NEM) Flip-Flops

The APE flip-flop is asserted when memory detects a parity error on the parity generated by Cache for RH70 address and control information. The RH70 sends address and control information to Cache, Cache computes parity and transfers the address, control, and parity to memory. Memory performs a parity check, and if a parity error is detected, sends a signal to Cache, which in turn causes Cache to assert the ADML ADRS PAR ERR H signal to the RH70.

Cache also sends a CCBD MBC TIMEOUT H signal which is applied to both the APE and NEM flip-flops. Since the TIMEOUT signal is applied to all four controllers, it is ANDed in gate E35-pin 3 with the proper controller select signal (CDPJ SEL-DATA CNTLX H). If the time-out occurs due to a non-existent memory, the NEM flip-flop is set and the APE flip-flop remains cleared. If a time-out occurs due to an address parity error, the APE flip-flop is set and the NEM flip-flop is inhibited from setting.

The APE error condition is bit 15 of the CS3 register and is ORed in gate E24-pin 12 to cause a parity error (PE bit 13 of CS2 register). The NEM error condition is bit 11 of the CS2 register. Both the APE and NEM flip-flops are cleared by the CSTB CLR ERR B L signal.

### 4.22 LOGIC DIAGRAM CSTD

This logic diagram contains the BUSI multiplexers for the CS1 and CS2 registers, and the clear logic used to store INIT or Program Clear conditions if they occur during a memory cycle.

#### 4.22.1 BUSI Multiplexers (CS1, CS2)

The BUSI multiplexers consist of four open-collector 8234 multiplexers used to gate out data from the CS1 and the CS2 registers when the program reads those registers.

BCTA CS1 OUT L, when asserted, enables the A0 – A3 (containing CS1 data) inputs to the multiplexer. BCTA CS2 OUT L enables the B0 – B3 (containing CS2 data) to the multiplexer. The BUSI outputs are applied to the Unibus as BUS DO L – D15 L.

### 4.22.2 INIT and Program Clear (PGCLR) Logic

If the programmer sets a 1 in the CLR bit position of the CS2 register, or if an INIT on the Unibus is raised due to a Reset instruction, the CSTD INIT.PGCLR flip-flop sets. This flip-flop stores the INIT or PGCLR condition in order to allow any memory cycle which might be in progress to complete, and to then prevent the next memory cycle from occuring. The CSTD PGCLR.INIT (0) signal from the flip-flop is applied to 50-ns delay line DL3 (see logic diagram CSTC) which causes CSTC DONE H to be asserted at the completion of any memory cycle in progress. This signal, coupled with the CSTD PGCLR.INIT signal, fires 400-ns one-shot E37-pin 4, causing CSTD CLR L and CSTD CLR H signals to be generated. These signals clear the RH70 and the associated drive. If the RH70 is in the RDY state and an INIT or PG CLR is asserted, 400-ns one-shot E37-pin 4 is fired immediately (due to RDY enabling pin 1 of E37), which causes the CSTD CLR L and CSTD CLR H signals to be generated.

The CSTD CLR L signal is also applied to discrete element Q1, resistor R15 and capacitor C67 to provide a small delay before clearing the CSTD PGCLR.INIT flip-flop.

If the RH70 is currently performing a memory cycle, the CSTA RDY B H and CSTC DONE H signals (both being unasserted) prevent the 400-ns one-shot from firing. When the memory cycle completes and DONE is asserted, or the RH70 goes to the Ready State, it will clear the RH70 and drive when an INIT or PGCLR condition occurs. A second 400-ns one-shot is shown (E37-pin 12). The function of this one-shot is to extend the period of the CSTA GO CLR pulse. GO CLR is asserted when a data transfer command is loaded in the drive. The output of the one-shot is ORed in NOR gate E32-pin 6 with BCTC DC LO L and the INIT.PGCLR output of one-shot E37-pin 4. The assertion of any of these signals asserts CSTD DB INIT L which initializes the data buffer logic (see logic diagram MDPA-MDPJ).

THE CSTA GO CLR L signal, in addition to being supplied to the 400-ns one-shot, is ORed in OR gate E4-pin 8 with the BCTC DC LO L and BCTC INIT.PGCLR to yield CSTD CLR.GO CLR L, which clears certain error and control bits in the RH70.

## 4.23 LOGIC DIAGRAM CSTE

This logic diagram contains the BUSI multiplexers for the data buffer and CS3 registers and the Start logic used to initiate Massbus cycles during write operations.

### 4.23.1 BUSI Multiplexers (DB, CS3)

BCTA CS3 OUT L, when asserted, enables the A0 - A3 inputs (containing data from the CS3 register) to the multiplexers. BCTA DB OUT L, when asserted, enables the B0 - B3 inputs to the multiplexers. These inputs contain data from OBUF in the data buffer. DB OUT L is asserted when the program attempts to read the data buffer during data buffer maintenance operation. The outputs of the multiplexers are supplied to the Unibus as BUSI D00 OUT L - BUSI D15 OUT L.

### 4.23.2 Four Counter

The four-count circuit counts memory cycles. After four counts, a carry is obtained which causes CSTE START H to be aserted. In the event that fewer than four memory cycles are to occur, AWRE WC OFLO (1) L is set, which also causes CSTE START H to be asserted. The START signal will cause the RUN line on the Massbus to be asserted. When the four counter overflows, the R3 output is fed back to the CUP (Countup) input via NAND gate E7-pin 8 to prevent the counter from counting higher than four.

The counter is loaded with the 2's complement of four (1100) when the CSTA GO CLR L signal is asserted. This signal is asserted when a data transfer command is loaded in the drive.

When the counter reaches a count of four, it indicates that four memory transfers have been initiated. This may mean that as many as eight data words are in the data buffer if all the transfers were double words. Typically, there will be eight words or seven words (allowing one single-word transfer) to put the next memory address on a double-word boundary.

## 4.24 M5904 MASSBUS TRANSCEIVER MBSA, MBSB, MBSC

The Massbus consists of three Massbus cables and associated Massbus transceiver modules. A 40-pin connector on each M5904 Massbus Transceiver module connects the transceivers to the Massbus cables.

Each signal on the Massbus is applied to a differential driver circuit (75113) which transmits the true signal and an inversion of the signal along the bus. At the other end of the bus, the signals are received by differential receivers (75107B) which output the true form of the signal. The differential circuitry serves to eliminate noise, since any common mode noise will be cancelled at the differential receivers. For additional description, refer to the M5904 Massbus Transceiver module description in this chapter.

The three Massbus cables are designated Massbus Cable A, Massbus Cable B, and Massbus Cable C (sheets 1,2,3 of drawing D-BS-RH70-0-1).

The M5904 Massbus Transceiver is functionally shown with the dotted block on each block schematic. The 40-pin connector is shown in the center of the dotted block. The differential transmitters which drive signals onto the Massbus from the RH70 are shown to the left of the connector. These signals originate at the drive and are routed to the RH70 via the differential receivers.

To minimize switching of signals on any transceiver module at a given time, the signals are grouped on different modules. For example, OBUF 00-05 H is contained on MBSA, OBUF 06-11 H is contained on MBSB and OBUF 12-15 H is contained on MBSC. The D00 IN H - D15 IN H signals from BUSA are also divided on the three modules in a similar manner. The RSEL 0 H - RSEL 4 H signals are grouped on MBSA and MBSB.

The OBUF signals are gated by GATE SYNCD H, which enables the output of OBUF to be gated onto the MASS "D" lines of the Massbus. The D00 IN H - D15 IN H signals, which form the MASS "C" lines are enabled by BCTB GATE CNTL H, which occurs when the RH70 is writing a remote register. GATE CNTL H is the assertion of BCTA DEV SEL H and BCTA CTOD H. The RSEL signals select a drive register and are enabled by the DEV SEL signal. Unit select signals U00 H - U02 H are also enabled by DEV SEL and specify one of eight possible drives. The remaining control signals which are supplied to the drive are also shown. These include WCLK, RUN, CTOD, CLR, AC LO, DEMAND (1), OBUF PA, CPA OUT.

The signals sent from the drive to the Massbus are SYNC D00 - SYNC D17, which represent synchronous data, and C00 H - C15 H, which represent the contents of a drive register. Control signals which include EXCP, EBL, ATTN, SYNC CLK, CPA IN, OCC and TRA are also shown.

## 4.25 M5904 MASSBUS TRANSCEIVER MODULE

The M5904 Massbus Transceiver module contains nine differential driver chips (75113) and seven differential receiver chips (75107B). Each driver chip and each receiver chip is capable of carrying two signals. Thus, the chips can be designated dual drivers and dual differential receivers. The transmission line connected to the transceivers are bidirectional in that they can both receive and transmit information. This is illustrated for one signal line in Figure 4-10.

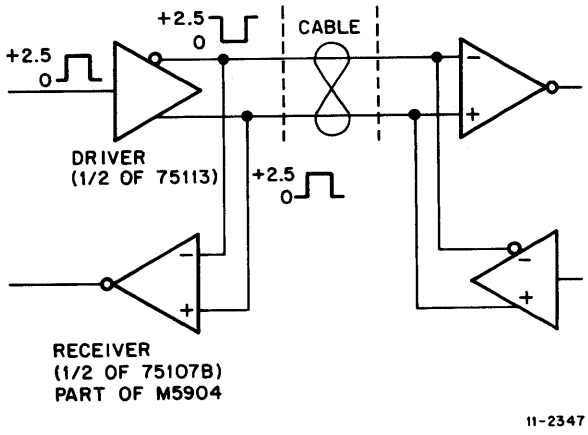



Figure 4-10 Typical Differential Driver/Receiver Connection

The advantage of differential circuitry is that any noise picked up is generally picked up on both the inverted and non-inverted signal lines. The differential receiver takes the difference between the signals, regardless of the noise level, and the noise is effectively cancelled out.

Each driver on the M5904 must be terminated since the M5904 is used to drive transmission lines (Figure 4-11).

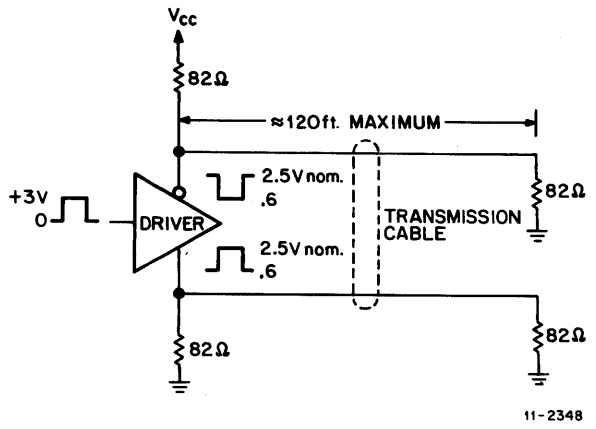



Figure 4-11 Driver Termination

The M5904 Massbus Transceiver requires input voltages of +5 Vdc and -15 Vdc. The dual drivers require +5 Vdc operating voltage while the dual differential receivers require +5 Vdc and -5 Vdc. The -5 Vdc is obtained from the -15 Vdc source via a resistor and Zener diode.

### 4.25.1 75113 Dual Differential Driver Chip

The 75113 Tri-State Dual Differential Driver Chips provide differential outputs with high current capability in order to drive balanced lines. The chips feature a high output impedance, making it possible to connect many drivers on the same transmission line. A simplified schematic of the 75113 is shown in Figure 4-12.

The inverting output of the driver chip is the transistor collector, while the non-inverting output is the transistor emitter, shown at point B. When the input is low, neither transistor conducts and line A is biased to +2.5 V, while line B is biased to 0 V by the terminator resistors (refer to diagram). When the input is high, the upper transistor collector is driven low (0 V) and the lower transistor emitter is driven high (+2.5 V). The pin connection diagram for the dual differential driver is shown in Figure 4-13.

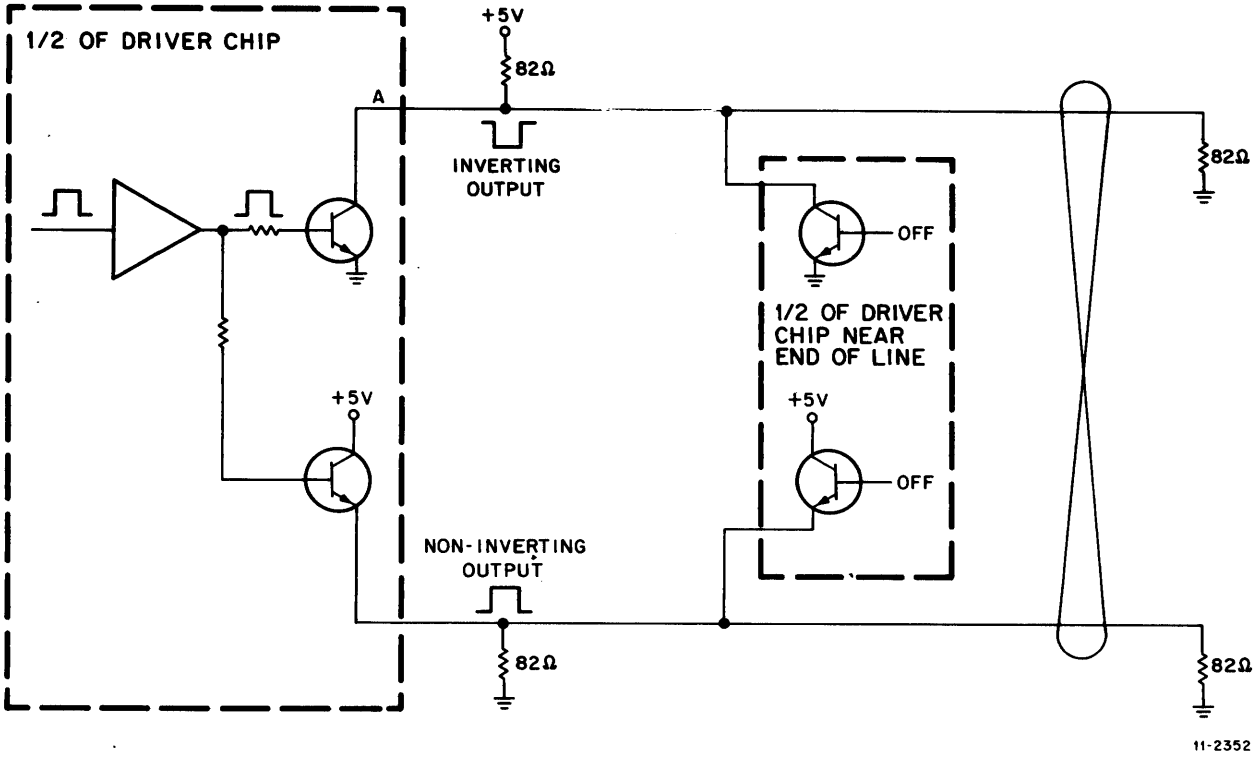



Figure 4-12 Driver Chip Simplified Schematic

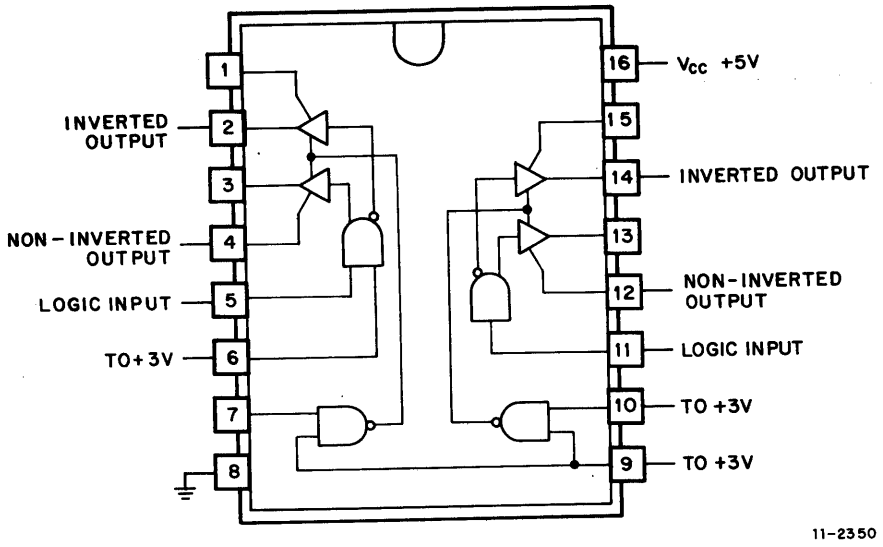



Figure 4-13 Dual Differential Driver Pin Connection Diagram

### 4.25.2 75107B Dual Differential Line Receiver Chips

The 75107B Differential Receiver Chips feature dual independent channels with common voltage supply and ground terminals. The circuits operate as follows. If the voltage at pin 1 is positive with respect to the voltage at pin 2, the output at pin 4 goes positive (Figure 4-14).

If the voltage at pin 1 is negative with respect to pin 2, the output at pin 4 goes negative. The pin connection diagram for the receiver is shown in Figure 4-15.

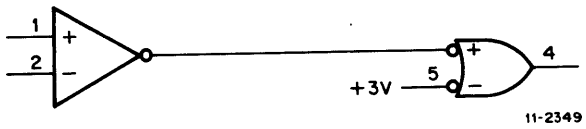



Figure 4-14 Simplified Line Receiver Logic Diagram

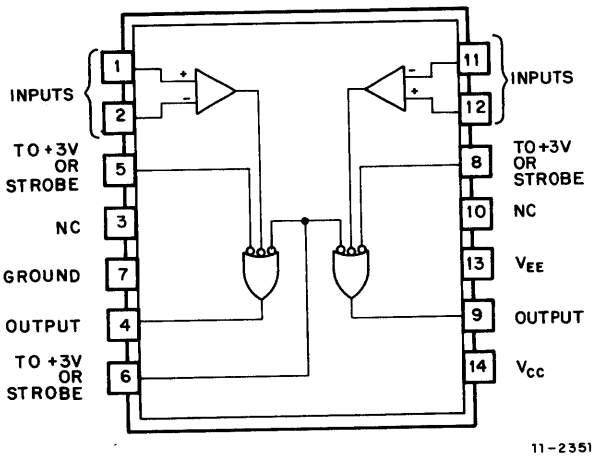



Figure 4-15 75107B Differential Receiver Pin Connection Diagram

## 4.26 H870 TERMINATOR

The H870 Bus Terminator provides a simple and reliable method of terminating the Massbus. The Massbus is terminated by plugging H870 terminators into each M5903 transceiver module in the last drive.

The H870 consists of 38 82 Ω -1/4 watt resistors wired between each Massbus line and a common ground connection.

**NOTE**

**The H870 terminators are to be installed on the M5903 transceiver module with the resistors facing up.**

# CHAPTER 5
# INSTALLATION AND MAINTENANCE

See the appropriate chapter in the accompanying Systems Manual (Chapter 2 in the **TWU45 Systems Manual**).